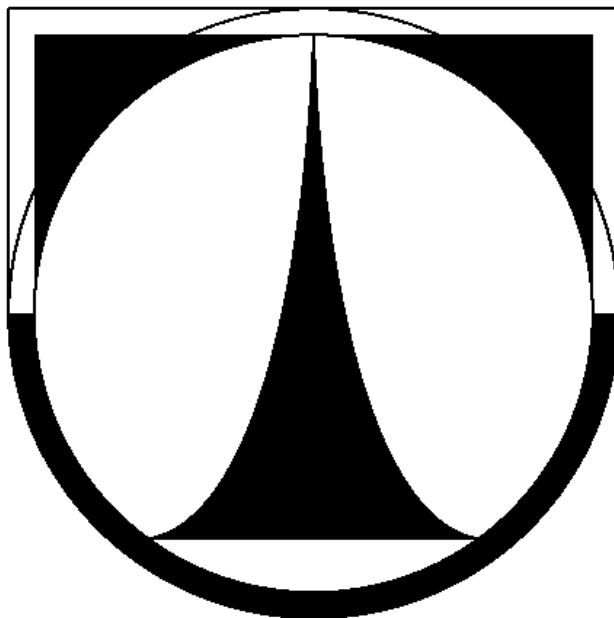


**TECHNICKÁ UNIVERZITA V LIBERCI**

**Ekonomická fakulta**



**BAKALÁŘSKÁ PRÁCE**

**2012**

**Kateřina Babcová**

# **TECHNICKÁ UNIVERZITA V LIBERCI**

## **Ekonomická fakulta**

Studijní program: B 6209 Systémové inženýrství a informatika

Studijní obor: Manažerská informatika

### **Metodika transformace aplikací pro řízení spolupráce do Web 2.0**

### **Methodology of transformation groupware applications to Web 2.0**

BP-EF-KIN-2012-01

Kateřina Babcová

Vedoucí práce: Mgr. Tomáš Žižka, katedra informatiky

Konzultant: Ing. Libor Hós, OR-CZ s.r.o.

Počet stran: 67

Počet příloh: 2

Datum odevzdání: 4. 5. 2012

## **Zadání ZP**

## **Zadání ZP**

# Prohlášení

Byla jsem seznámena s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. O právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracovala samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci, 4. 5. 2012

# **Anotace**

Tato bakalářská práce se zabývá vývojem aplikací na platformě Lotus Notes/Domino. Převážně vycházím ze zkušeností nasbíraných při mé roční řízené praxi ve společnosti OR-CZ spol. s r.o. V první části představím produkty Lotus Notes/Domino, Domino Designer a novou technologii XPages, umožňující tvorbu moderních webových aplikací. Následuje porovnání způsobu vývoje klasických Lotus Notes aplikací oproti webovým se zdůrazněním největších rozdílů. Další část práce už se zaměřuje na tvorbu webových aplikací pomocí XPages. Pro jejich vývoj se využívá JavaScriptu, XML, jazyka funkcí, HTML a kaskádových stylů. Ukázán je praktický převod stávající aplikace do formy pro čtení a vytvoření nové aplikace. Na konci jsou shrnuty základní výhody i nevýhody XPages.

## **Klíčová slova**

Lotus Notes/Domino, Domino Designer, XPages, JavaScript, webová aplikace, klientská aplikace

# **Anotace**

This bachelor's dissertation deals with applications development on Lotus Notes/Domino platform. It is largely based on my experiences gained during long-term practice at OR-CZ Ltd. In first part I will introduce Lotus Notes/Domino products, Domino Designer and new technology called XPages, which allows creation of modern web applications. Next part focus on difference between development of classical Lotus Notes applications and web applications followed by chapter dedicated to web application development using XPages. JavaScript, XML, Formula language, HTML and CSS (cascaded style sheets) are used for their development. Redesign of existing application to read only form and new application development is shown. At the conclusion summary of pros and cons is presented.

## **Key words**

Lotus Notes/Domino, Domino Designer, XPages, JavaScript, web application, client application

# Obsah

<b>Seznam obrázků.....</b>	<b>10</b>
<b>Seznam zdrojových kódů .....</b>	<b>11</b>
<b>Seznam použitých zkratk .....</b>	<b>12</b>
<b>Úvod .....</b>	<b>13</b>
<b>1 Představení IBM Lotus a XPages.....</b>	<b>14</b>
1.1 Lotus Notes .....	14
1.1.1 Zabezpečení .....	15
1.1.2 Nejnovější verze Lotus Notes .....	16
1.2 Domino Designer .....	17
1.3 XPages .....	18
1.3.1 Technologie .....	19
<b>2 Rozdíly při vývoji pro klienta a webové prostředí .....</b>	<b>21</b>
2.1 Vývoj pro klienta Lotus Notes .....	22
2.1.1 Popis elementů návrhu.....	22
2.1.2 Vývojová perspektiva .....	23
2.1.3 Programovací jazyky .....	24
2.2 Vývoj pomocí XPages .....	25
2.2.1 Vývojová perspektiva .....	26
2.2.2 Programovací jazyk .....	28
2.3 Rozdíly při vývoji XPages a klientských aplikací Lotus Notes .....	28
2.3.1 Formulář .....	28
2.3.2 Rozložení prvků v aplikaci .....	29
2.3.3 Navigace .....	30
2.3.4 Ladění aplikace.....	30
2.3.5 Rozdílné přihlašování .....	32
<b>3 Převod stávajících aplikací a tvorba nových.....</b>	<b>33</b>
3.1 Tvorba nové aplikace.....	33
3.1.1 Postup tvorby nové aplikace.....	33



3.2	Převod stávajících aplikací .....	34
3.2.1	Postup převodu aplikace již vytvořené pro klienta.....	34
3.3	Aplikace Obchodní aktivity .....	35
3.3.1	Úvodní stránka.....	36
3.3.2	Rozložení Úvodní stránky .....	37
3.3.3	Navigace .....	38
3.3.4	Obchodní případ .....	39
3.3.5	Panel akcí.....	40
3.3.6	Hlavička.....	41
3.3.7	Obsah obchodního případu.....	41
3.3.8	Záhlaví obchodního případu.....	45
3.3.9	Pohledy .....	45
<b>4</b>	<b>Ukázka vývoje aplikace.....</b>	<b>48</b>
4.1	Vytvoření úvodní stránky .....	48
4.2	Profil .....	48
4.2.1	Ověřování zadaných hodnot .....	50
4.2.2	Dojo .....	51
4.2.3	Type Ahead.....	52
4.3	Pohledy .....	52
<b>5</b>	<b>Zhodnocení.....</b>	<b>55</b>
5.1	Přínosy XPages .....	55
5.2	Nevýhody.....	57
	<b>Závěr .....</b>	<b>59</b>
	<b>Citace .....</b>	<b>60</b>
	<b>Bibliografie .....</b>	<b>61</b>
	<b>Seznam příloh .....</b>	<b>62</b>

## Seznam obrázků

Obr. 1: Perspektiva pro vývoj klientských aplikací.....	24
Obr. 2: Perspektiva pro vývoj XPages.....	26
Obr. 3: Rozložení rámců.....	30
Obr. 4: Debugger pro LotusScript .....	31
Obr. 5: Source .....	37
Obr. 6: Zobrazení XPage místo formuláře .....	40
Obr. 7: Nastavení jednoduché akce .....	41
Obr. 8: Datový zdroj XPage .....	42
Obr. 9: Kontakty .....	43
Obr. 10: File download.....	44
Obr. 11: Zobrazení ikon ve sloupci .....	47
Obr. 12: Validace emailu.....	51
Obr. 13: Type Ahead pro pole město .....	52
Obr. 14: Pohled v Domino Designeru .....	53
Obr. 15: Vzhled starší webové aplikace .....	56
Obr. 16: Ukázková aplikace vytvořená pomocí XPages .....	56

## Seznam zdrojových kódů

Ukázka kódu 1: Jazyk funkcí.....	25
Ukázka kódu 2: LotusScrip.....	25
Ukázka kódu 3: Kaskádové styly .....	38
Ukázka kódu 4: XML kód pro pozadí v banneru .....	38
Ukázka kódu 5: JavaScript pro link.....	38
Ukázka kódu 6: JavaScript pro include page.....	39
Ukázka kódu 7: JavaScript pro pole cena.....	42
Ukázka kódu 8: Jazyk funkcí pro pole firma.....	43
Ukázka kódu 9: JavaScript pro pole firma .....	44
Ukázka kódu 10: Kód pro filtraci pohledu .....	45
Ukázka kódu 11: Vypočítávané ikony.....	46
Ukázka kódu 12: Vypočítávaný sloupec s odpovědným.....	47
Ukázka kódu 13: XML pro styl tabulky .....	49
Ukázka kódu 14: Ukázka použitých CSS.....	49
Ukázka kódu 15: JavaScript pro vypočítávané pole.....	49
Ukázka kódu 16: Viditelnost tlačítka editovat .....	50
Ukázka kódu 17: Viditelnost tlačítka uložit .....	50
Ukázka kódu 18: Regulérní Výraz pro email .....	50
Ukázka kódu 19: Cesta k poli s městy.....	52
Ukázka kódu 20: Výběr dokumentů pro pohled.....	53

## Seznam použitých zkratek

ACL	Acces control list (seznam přístupových práv)
AJAX	Asynchronous JavaScript and XML
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
IDE	Integrated development environment
JSF	Java Server Faces
JSP	Java Server Pages
OA	Obchodní aktivita
RAD	Rapid Application Development
XML	eXtensible Markup Language
WYSIWYG	What you see is what you get

# Úvod

Celá bakalářská práce vzniká na základě roční řízené praxe, kterou vykonávám ve firmě OR-CZ spol. s r. o. Nastoupila jsem do oddělení Lotus Notes, které se zabývá vývojem a úpravou aplikací na platformě Lotus/Notes Domino. Společně s mým garantem jsme vybrali téma a zároveň i náplň praxe, tj. vývoj a údržba aplikací Lotus Notes klasickým způsobem a pomocí XPages. XPages je nový designérský prvek umožňující tvorbu webových aplikací.

Webové aplikace jsou stále oblíbenější. Souvisí to hlavně s rozvojem internetu, ke kterému se dá v dnešní době připojit téměř kdekoliv. Díky novým webovým technologiím lze vytvářet webové aplikace, které se funkčností téměř vyrovnávají těm klasickým. Proto i Lotus Notes/Domino přišlo s novým prvkem XPages, který za pomoci základních i moderních technologií, umožňuje vytvářet podnikové webové aplikace. Než se začne jakákoliv aplikace vyvíjet je potřeba zvážit, zda bude pro klienta či web. Každý způsob vývoje s sebou nese své výhody a nevýhody. Pojem aplikace pro klienta v práci budu používat pro odlišení od webové aplikace. Obvykle se pro toto odlišení využívá označení desktopová aplikace. Poněvadž Lotus Notes/Domino je architektura typu klient/server, raději v práci budu používat pojem klientská aplikace. U desktopové aplikace se obvykle počítá s tím, že pracuje s daty uloženými na lokálním disku, což u Lotus Notes/Domino neplatí, jelikož aplikace jsou uloženy na serveru a uživatel k nim přistupuje pomocí klienta Lotus Notes.

Cílem mé práce je zhodnotit vývoj webových aplikací pomocí XPages. Porovnávat ho budu se způsobem vývoje klientských aplikací. V praktické části bude ukázán převod aplikace do XPages a také vytváření zcela nové aplikace. Na konci práce shrnu výhody a nevýhody vývoje pomocí Xpages.

# 1 Představení IBM Lotus a XPages

IBM Lotus software představuje skupinu produktů, které slouží k podpoře spolupráce, komunikace ve firmách všech velikostí. Zde je uvedeno několik produktů, které IBM aktuálně nabízí jako součást Lotusu [1]:

- Lotus Notes/Domino
- Lotus Domino Designer
- Lotus Sametime
- Lotus iNotes
- IBM Connections
- Lotus Symphony
- Lotus Notes Traveler
- IBM Web Content Manager
- IBM WebSphere Portál

K používání a vývoji Lotus Notes aplikací je potřeba mít k dispozici 4 produkty, a to: Lotus Domino, Lotus Administrátor, Lotus Domino Designer a Lotus Notes. Lotus Notes a Lotus Domino společně tvoří architekturu typu klient/server. Zkráceně se proto uvádí název Lotus Notes/Domino. Aplikace v této architektuře je umístěna na serveru a klient komunikuje s touto aplikací. Klient se nazývá Lotus Notes a serverová část Lotus Domino. Ovládání Domino Serveru se uskutečňuje pomocí další aplikace: Lotus Administrátor. Vytvářejí se zde ID pro uživatele a zároveň se jim přidávají či odebírají přístupová práva. Spravují se tady skupiny, servery, opravují poškozené databáze apod.

## 1.1 Lotus Notes

Lotus Notes je „podnikové e-mail řešení, které sdružuje v jednom jediném prostředí přístup a možnost komunikace s dalšími lidmi, podnikovými aplikacemi a jinými zdroji v rámci podniku.[2, s. 18]

Základem je poštovní schránka, kalendář, kontakty, osobní úkoly a rezervace prostředků. Poskytuje také platformu pro vlastní vývoj nových aplikací. Aplikace mohou být řízeny

pomocí workflow, které automatizuje celou řadu procesů. „*Správně nastavené workflow dokáže podpořit zjednodušení a zrychlení jednotlivých činností nebo odstranění schvalovacího procesu prostřednictvím kolujícího papíru, o němž často ani nevíme, ve které kanceláři zrovna leží.*“ [3, s. 32] Všechny procesy, které se týkají schvalování, postupování, připomínkování a jiné, zajišťuje právě workflow. Vždy, když je v některém z procesů potřeba vaše účast, přijde vám o tom e-mail. Například, když si budete žádat o dovolenou. Vyplníte elektronickou žádanku. Aplikace automaticky pošle vaši žádost nadřízenému a v momentě, kdy on ji schválí, přijde vám potvrzení emailem. Workflow tak vede k efektivnější a rychlejší spolupráci.

### **1.1.1 Zabezpečení**

Zásadní vlastností systému Lotus Notes/Domino je zabezpečení. „*Na rozdíl od některých ostatních aplikací, u nichž je zabezpečení řešeno až dodatečně, systém Lotus Notes/Domino je od počátku stavěn na velmi kvalitní bezpečnostní architektuře.*“ [4, s. 171] Zajišťuje to, že informace jsou přístupné pouze těm, komu jsou určeny. Zabezpečení začíná u serveru, přes přístup k jednotlivým aplikacím až po skrývání jednotlivých polí na dokumentu.

Pro přístup k serveru se využívá adresáře systému Domino, kde je určeno kdo smí a nesmí k serveru přistupovat, a k čemu mají oprávnění. Další úroveň je zabezpečení jednotlivých aplikací. [4]

U každé aplikace zvlášť se dají nastavit různé úrovně přístupových práv. Jednotlivé úrovně jsou vidět v ACL (Access Control List). Zároveň se v něm i nastavují. Lze nastavit nejen přístupy jednotlivých uživatelů, ale i skupin. Úrovní je celkem sedm a jsou seřazené od nejdůležitější po nejnižší. Nejvyšší úrovní je manažer, který má veškerá práva a může přiřazovat přístupová práva dalším uživatelům. Následuje návrhář, který upravuje návrh aplikace, ale nemůže mazat stávající dokumenty v aplikaci. Dále editor, který může editovat libovolný dokument. Autor má možnost číst, vytvářet dokumenty a smí upravovat pouze dokumenty, ke kterým má přístup. Čtenář má dovoleno číst obsah databáze, ale nesmí nic měnit. Zapisovatel má umožněno vytvářet dokumenty, ale nesmí číst obsah aplikace. Posledním je bez přístupu, tato úroveň má zakázaný jakýkoliv přístup k aplikaci.

Kromě toho se v ACL přiřazují uživatelům jednotlivé role, které obvykle souvisejí s funkcemi ve firmě. Role jsou podobné jako úrovně, ale určují i přístup k jednotlivým dokumentům, polím a vytvářejí se pro každou aplikaci zvlášť. [4]

Pro přístup k jednotlivým dokumentům se nejčastěji využívá polí *authors* a *readers*. Ty určují, kdo bude smět dokument číst, nebo upravovat. Doporučuje se u každé aplikace založit roli, která bude vždy v polích *authors* a *readers*, aby v případě potřeby byl stanoven někdo, kdo bude mít přístup ke všem dokumentům. Taková struktura přístupových práv vytváří vysokou úroveň zabezpečení podnikových informací. [4]

### 1.1.2 Nejnovější verze Lotus Notes

První verze 1.0 se objevila roku 1989. Obsahovala elektronickou poštu, posloupnost diskuzí, zabezpečení, podporu sítí, ale i šablony pro tvorbu vlastních aplikací. [5] Vývoj postupoval přes několik verzí. Přidávali se různé funkcionality, nové prvky, vzhled a další. Nyní je nejaktuálnější verze 8.5.3.

Verze 8.5.3 byla vydána v říjnu 2011. Od verze 8, z roku 2007, přináší spoustu novinek. [6] Na první pohled je viditelný nový moderní vzhled, který se v mnohém podobá Microsoft softwaru. Zejména podoba a ovládání poštovní schránky je hodně obdobné jako u Microsoft Office Outlook.

Největším přínosem pro vývojáře je jádro plného klienta Lotus Notes založené na otevřené platformě Eclipse. „*Eclipse provides kind of universal tool platform - an open extensible IDE for anything and yet nothing in particular.*“ V češtině: Eclipse poskytuje univerzální druh platformy - otevřené rozšiřitelné vývojové prostředí na všechno a přesto na nic konkrétního.[7] Eclipse tak poskytuje základní framework, který je dále rozšiřitelný v podobně plug-in. Díky tomu například přibyli v Domino Designeru editory pro HTML, CSS, XML.

Podle IBM „*The transition to Eclipse has yielded benefits without incurring any losses.*“ V českém překladu: Přechod na Eclipse přináší spoustu novinek, aniž by vznikly nějaké ztráty. [6] Také aplikace, které běžely pod starší verzí, zde fungují.



Novinkou od verze 8.5 je technologie XPages, o které se podrobněji píše dále v textu.

## 1.2 Domino Designer

Domino Designer je nástroj pro vytváření a přizpůsobení aplikací, které mohou být spouštěny skrze klienta Lotus Notes, webový prohlížeč nebo mobilní zařízení. Začleňuje se do skupiny RAD (Rapid Application Development), neboli „Rychlý vývoj aplikace.“ Obsahuje nástroje, které umožňují vyvíjet grafické prvky způsobem „Drag and Drop“, protože podporuje WYSIWYG. Je to zkratka pro „What you see is what you get“, česky „co vidíš, to dostaneš“. Díky tomu výsledná stránka v Domino Designeru je velice podobná finálnímu vzhledu stránky.

Stejně jako klient Lotus Notes je Domino Designer od verze 8 začleněn do platformy Eclipse. Základem prostředí Eclipse je práce v editorech a pohledech. V pohledech se zobrazují informace o otevřených aplikacích a vložených prvcích. V editorech se odehrává všechno nastavování a programování. Tyto pohledy a editory lze přesouvat, minimalizovat, či zvětšit na celou plochu. Rozložení oken společně vytváří perspektivy. [8]

K dispozici jsou čtyři základní perspektivy:

- Domino Designer: obsahuje všechny základní elementy
- XPages: obsahuje okna nezbytná pro vývoj XPages
- Forms/Views: neobsahuje elementy, které jsou pouze pro XPages
- Java: pro programování v Javě.

Tyto perspektivy se dají dále přizpůsobit anebo vytvářet vlastní.

Programování se uskutečňuje pomocí několika různých jazyků zahrnujících jazyk vzorců, LotusScript, JavaScript, Java. Není potřeba je všechny ovládat. Nejvyužívanějšími je LotusScript, @Funkce a JavaScript pro vývoj webových aplikací.

Každá aplikace je tvořena daty a dalšími elementy návrhu. Do dat se ukládají obrázky a knihovny. Elementy návrhu jsou právě ty části návrhu, které se využívají k tvorbě aplikací.[4]

Elementy návrhu:

- Formuláře
- Pohledy
- Rámce
- Stránky
- Sdílené prvky: podformuláře, pole, sloupce, osnovy
- Komponenty složené aplikace
- Agenti
- XPages
- Custom Controls

## 1.3 XPages

Byly oficiálně vydané v lednu 2009, společně s novou verzí Lotus Notes/Domino 8.5. Ve stejný měsíc se konala pravidelná konference Lotusphere v Orlandu, kde byli XPages ve spoustě prezentací a seminářů představeny. *“It was variously described as a new framework for Web 2.0 development, a strategic move to reinvigorate the application-development experience, a standards-based runtime that would greatly boost productivity for the Domino web developer etc.”* V češtině: Byly různě představeny jako nový framework pro vývoj Web 2.0, strategický krok k oživení vývoje aplikací, na standardech založený runtime, který zvýší produktivitu Domino web vývojářů. [9]

XPages je nový designérský prvek v Domino Designeru, který umožňuje tvorbu moderních webových aplikací. Moderní hlavně proto, že se pomocí nich vytváří Web 2.0 aplikace. Web 2.0 je *„krátké a všeobecně přijímané označení pro web tvořený běžnými uživateli.“* [10, s. 4] Nahradil se tak statický obsah stránek za prostor pro sdílení informací.

Od verze 8.5.1 je k dispozici nová technologie, která umožňuje provoz v klientu Lotus Notes. Znamená to, že lze aplikaci vytvořit jednou a bude fungovat v prohlížeči i klientu. Pomocí XPages se dají také vytvořit aplikace pro mobilní zařízení.

### 1.3.1 Technologie

Celé XPages jsou postaveny na Javě - JavaServerFaces, ale vývojář ji k vývoji nutně nepotřebují umět. Při dobré znalosti se ovšem dá využít jako velmi silný vývojový nástroj. Ve skutečnosti existence JSF jako základní vrstva XPages je kompletně skrytá a pro velkou většinu aplikací je nepotřebná.

Výchozí vrstva pro JSF je poskytnuta prostřednictvím technologie známé jako JSP. Tato vrstva byla nahrazena XML, které velmi zjednodušuje programovací model. [9]

XML je zkratka pro eXtensible Markup Language. Přeloženo do češtiny to znamená rozšiřitelný značkovací jazyk. Z názvu tedy vyplývá, že patří do skupiny značkových jazyků. Tyto technologie umožňují pomocí značek, neboli tagů, označovat části textu. XML je velice podobný HTML, ale je tu několik zásadních rozdílů. Jazyk HTML se stal jedním z nejoblíbenějších jazyků pro tvorbu webových stránek, a to hlavně díky své jednoduchosti. Čím byl jazyk používanější, tak postupně rozšiřovalo množství značek a ztrácela se jeho jednoduchost. Vznikl tak požadavek na jazyk, který bude mít menší počet značek a zároveň mít možnost téměř neomezeného množství značek. Tak vznikl nový značkovací jazyk XML, jehož základní vlastností je, že nemá žádné předem určené tagy. Poskytuje možnost vytvářet si tagy vlastní. Zároveň má jednoduchou syntaxi, ale vyžaduje striktní dodržování těchto pravidel. [11]

Přesto výchozí programovací rozhraní není XML. Výchozí pohled v Domino Designeru nabízí vývojářům interaktivně umisťovat ovladače a objekty pomocí uchycením myši a nastavovat vlastnosti vyplněním, či vybráním hodnoty z panelu vlastností. Všechny tyto aktivity generují odpovídající XML značky, které lze zobrazit či editovat skrze speciální editor v Domino Designeru. [9]

Vývoj XPages je řízen kombinací CSS, HTML, JavaScriptu a XML. Vzhled je určován pomocí CSS, nebo přes vlastnost styl, kde se dá určit pro každý prvek zvlášť. Je tu i další nový prvek, a to témata, což jsou šablony pro vzhled.

Moderní technologie, které XPages podporují, jsou Dojo a AJAX. Dojo je JavaScriptová knihovna, nabízející prvky a funkcionality, které mohou být použity na webové stránky. Díky tomu se dají vytvářet rich text editory nebo malý kalendář pro výběr data. Nativní

podpora AJAX (Asynchronous JavaScript and XML) umožňuje JavaScriptu na straně klienta volat v pozadí server a podle potřeby získávat určitá data. V XPages tak umožňuje vytvářet type-ahead (což je průběžné hledání při psaní) a částečnou aktualizaci stránek. [12]

Základní technologie pro XPages tedy jsou:

- HTML
- XML
- CSS (Cascading Style Sheets)
- JavaScript
- @Funkce
- AJAX (Asynchronous JavaScript and XML)
- Dojo
- JSF (JavaServer Faces) [13]

## 2 Rozdíly při vývoji pro klienta a webové prostředí

Webové aplikace se nejvíce liší od aplikací typu klient/server tím, že běží na serverech umístěných na internetu, namísto na serverech ve firmě a uživatelé k těmto aplikacím přistupují přes internet a jako klient se používá webový prohlížeč. V případě Lotus Notes je to ještě trochu jinak. Klientské i webové aplikace běží na serverech ve firmě.

Webové aplikace jsou stále využívanější, a to hlavně díky rozšíření internetu a s ním spojený rozvoj webových technologií. Původně byli webové stránky pouze prostředníky pro sdělování a šíření informací. Nyní se přechází k interaktivním webovým aplikacím, které nabízejí spousty funkcionalit, a dokážou se v mnohém rovnat možnostem klientských aplikací. Webové aplikace se využívají pro mnoho podnikových informačních systémů, internetových obchodů, sociálních sítí.

Přenos uživatelského rozhraní na web je často problematický. Uživatelé očekávají, že aplikace se bude chovat stejně, jako když ji spouštějí skrze klienta. Dále je problémem, že webové prohlížeče jsou různé a nechovají se všechny stejně, proto je potřeba testovat alespoň na několika prohlížečích. Přesto jsou webové aplikace oblíbené hlavně díky těmto výhodám[12]:

- **Jednoduché a levné dodání uživatelům.** Díky snížení nákladů spojených s instalací klientů. Kromě instalace prohlížeče nevyžadují žádné další instalace. Navíc prohlížeče bývají často součástí operačních systémů.
- **Jednoduché a levné aktualizace.** Pro aktualizaci klienta je potřeba provést jeho opětovnou instalaci. To v případě používání aplikací přes webový prohlížeč odpadá.
- **Lepší dostupnost aplikací.** Webová aplikace se může prostřednictvím internetu spustit na libovolném počítači přes webový prohlížeč. Lze tak pohodlně pracovat z domu, nebo jiného pracoviště.
- **Webové aplikace nejsou závislé na operačním systému počítače,** což u klienta často bývá problémem. Je jen třeba dát si pozor na jednotlivou kompatibilitu

s různými druhy a verzemi webových prohlížečů. V případě problémů je ale mnohem snazší instalovat si nový prohlížeč než celý operační systém.

- Tyto aplikace také mohou být spuštěny přes prohlížeč v **tablet**u nebo **mob**ilním **tele**fonu.

## 2.1 Vývoj pro klienta Lotus Notes

Pro vývoj klientských aplikací se využívá Domino Designer. Základní elementy, které se obvykle objevují v každé aplikaci, jsou:

- Formuláře
- Pohledy
- Rámce
- Podformuláře
- Osnova
- Stránky
- Agenti

### 2.1.1 Popis elementů návrhu

Formulář je základní element návrhu. Slouží pro zadávání a zobrazování dat v dokumentech. Může se skládat z tabulek, tlačítek, textu, polí, podformulářů, obrázků, vypočítávaného textu, vložených pohledů. Vložené pole může být několika typů - textové, číselné, s datem, zaškrťovací políčko, rich text pole, výběrové pole a další.

Nezbytným prvkem pro každou aplikaci jsou pohledy. Pohled se sestává z jednotlivých sloupců a řádků. Každý řádek představuje jeden dokument. Sloupce obsahují údaje o dokumentech, jako název, autor, datum vydání a podobně. Dle sloupců se dají dokumenty seřadit vzestupně či sestupně a také seskupovat.

Dalším prvkem návrhu jsou rámce. Rámce se používají pro rozvržení struktury aplikace. Rámec je jedna oblast, panel, nebo celé okno. Vkládají se do něj formuláře, pohledy,

stránky nebo další rámce. Tím se vytváří struktura jako levý postranní panel, hlavička patička.

Ve sdílených prvcích se nacházejí sdílená pole, podformuláře, osnova. Jsou to prvky, které se v aplikaci používají opakovaně. Stačí je navrhnout pouze jednou a pak už jen vkládat, kam je potřeba. Jejich hlavní výhodou je, že je lze upravit na jednom místě a změna se to na všech výskytech. Podobným prvkem jsou i sdílené kódy. Kam se ukládají často používané kódy jako zavření dokumentu, otevření, uložení a další.

Osnova je důležitou součástí každé aplikace, jelikož se pomocí ní vytváří navigace aplikace. Vytváří se automaticky na základě vytvořených pohledů. Obsahuje jednotlivé řádky jako odkazy na prvky, nebo mohou být seskupené do složek.

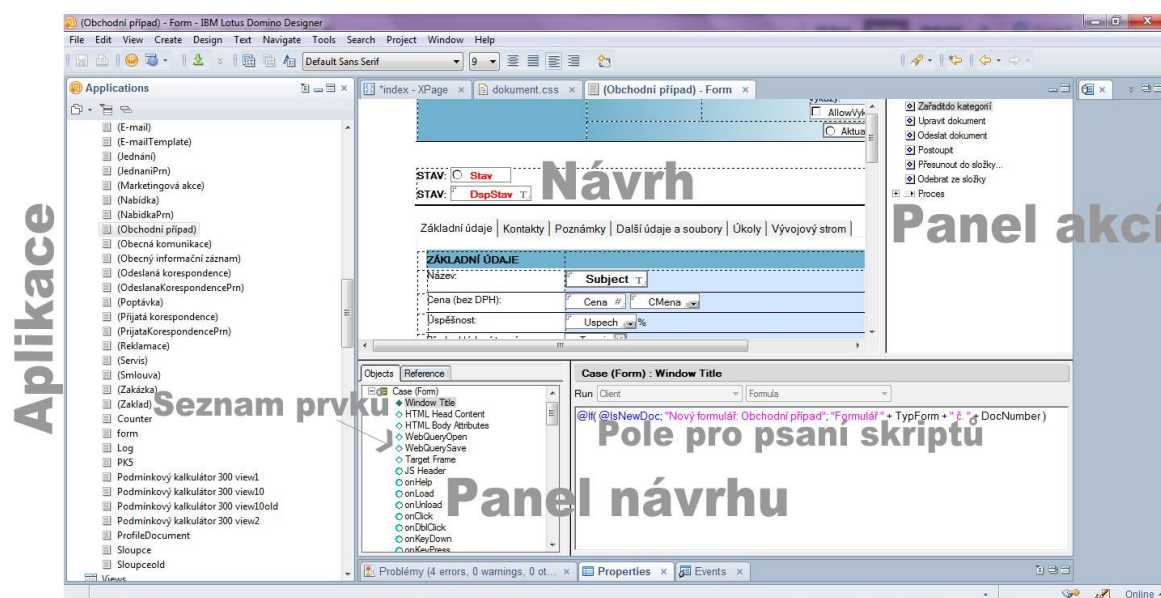
Kromě již zmíněných prvků, aplikace často obsahují agenty. Agent je kód běžící v pozadí aplikace. Je spouštěn pravidelně v nastavený čas (to bývá většinou v noci, aby běžící agenti v pracovní době nezatěžovali server) nebo je spouštěn uživatelem. Příkladem agenta je třeba narozeninový agent. Ten pravidelně každý den ve dvě hodiny ráno projede adresář s daty narození. Pokud najde shodu s aktuálním datem, oslavenci pošle e-mail a ostatním uživatelům upozornění, aby dotyčnému nezapomněli popřát k narozeninám.

### **2.1.2 Vývojová perspektiva**

Pro vývoj v Domino Designeru se využívá těchto oken:

- Aplikace
- Návrh
- Panel akcí
- Seznam objektů
- Programovací panel

Jejich výchozí rozložení, neboli perspektiva, je vidět na *Obr. 1: Perspektiva pro vývoj klientských aplikací*.



Obr. 1: Perspektiva pro vývoj klientských aplikací

zdroj: vlastní printscreen aplikace Domino Designer

Vlevo je seznam aplikací. Při rozbalení aplikace je vidět soupis elementů návrhu, které jsou seřazené podle typů. Jednotlivé aplikace mohou být přidávány, otevírány, odstraněny skrze kliknutí pravým tlačítkem.

V panelu akcí se vytvářejí tlačítka pro formulář. Pod nimi je ukrytý kód pro jakoukoliv akci. Tyto akce mohou být sdílené, pokud je využíváme na více formulářích.

Panel návrhu se skládá ze seznamu prvků návrhu a pole pro psaní skriptů, či funkcí. Prvky jsou hierarchicky seřazené. U každého prvku jsou vypsány jeho možné události a vlastnosti, které lze naprogramovat. Událost vznikne tehdy, když nějaký prvek, třeba formulář, něco udělá. Pro formuláře jsou typické události jako zavření, otevření, uložení. Tyto události se dělí, jestli vzniknou před (Query) anebo po (Post) vzniku události. Například QuerySave se spustí před uložením a PostSave po uložení formuláře. [4]

### 2.1.3 Programovací jazyky

Základními programovacími jazyky je LotusScript a jazyk vzorců. Jazyk vzorců je prvním programovacím jazykem Lotus Notes. Všechny funkce v jazyce vzorců začínají @. Příkladem je třeba @Command, @Prompt, @If a další. LotusScript se prvně objevil až u



verze IBM Lotus Notes 4. Je to objektově orientovaný jazyk podobný Visual Basicu, který významně překračuje možnosti @Funkcí. [14]

Pro porovnání obou jazyků se vytvořil příklad. Téměř každá učebnice o programování začíná příkladem s vypsáním textu „Ahoj světe“. Proto se použije stejný příklad i zde, pouze s rozdílem, že pozdrav se zobrazí až po kliknutí na tlačítko. První příklad je v jazyce vzorců a druhý v LotusScriptu. Výsledek je v obou případech stejný.

```
@Prompt ([Ok]; "Pozdrav"; "Ahoj světe")
```

*Ukázka kódu 1: Jazyk funkcí*

```
Sub Click(Source As Button)  
    result = MessageBox ("Ahoj světe", 0, "Pozdrav")  
End Sub
```

*Ukázka kódu 2: LotusScript*

V některých případech si lze vybrat, zda použít @Funkce, nebo LotusScript, ale ne vždy jde LotusScript použít, nebo je příkaz pomocí @Funkcí mnohem jednodušší. Například vzorec pro skrývání jednotlivých prvků v Domino Designeru nelze napsat v LotusScriptu. V LotusScriptu dokonce nejsou zastoupeny funkce. Například @DbLookup a DbColumn, které slouží k rychlému vyhledávání a načítání dat z aplikací.

## 2.2 Vývoj pomocí XPages

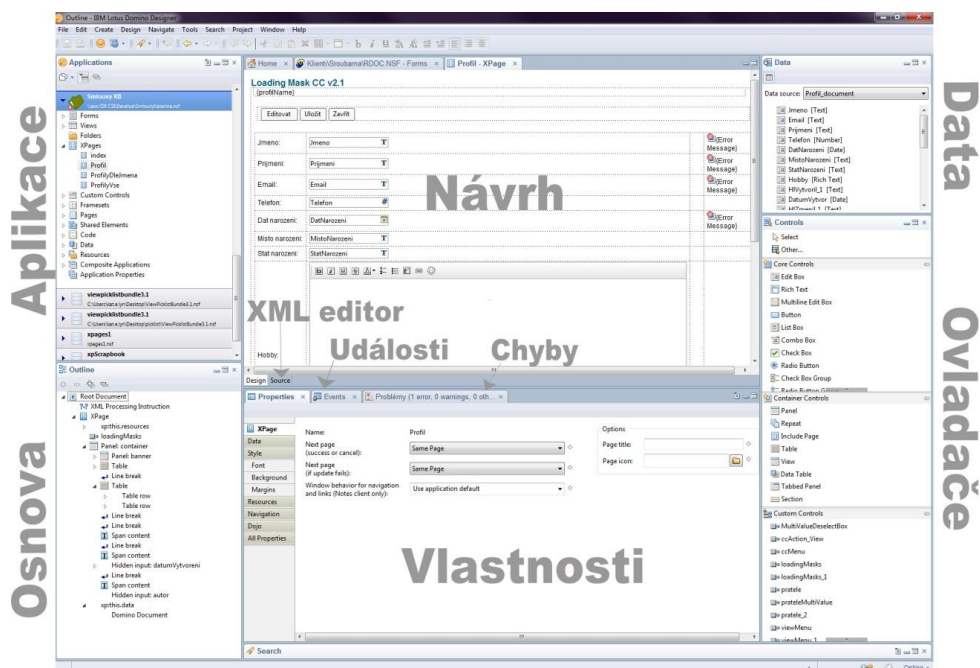
Pro vývoj XPage již není potřeba využívat rámců, osnov, stránek, ale stále je potřeba pracovat s formuláři a pohledy. Dále přibýly nové prvky, a to XPages a Custom Controls. XPage je podobná formuláři. Pomocí „Drag and Drop“ na ní lze umisťovat jednotlivé ovladače. Ovladače se dělí na tři skupiny: základní ovládací prvky, kontejnerové ovladače a Custom Controls. Základní ovládací prvky jsou například textová pole, popisky, tlačítka, rich text pole, zaškrtačovací pole, link, obrázek. Kontejnerové prvky obsahují strukturované prvky, pomocí kterých se vytváří rozložení stránky. Zahrnují panel, tabulku, pohled, sekce a jiné. Custom Controls jsou vlastní navržené ovládací prvky. Skládají z kontejnerových a základních ovládacích prvků. Jsou podobné podformulářům, využívají se zejména pro části

návrhu, které se vyskytují opakovaně. V Custom Controls je jednou navrhne a můžeme je použít na několika místech [13].

## 2.2.1 Vývojová perspektiva

Perspektiva (zobrazena na obrázku *Obr. 2: Perspektiva pro vývoj XPages*) pro vývoj pomocí XPages v Domino Designeru je složena z těchto oken:

- Aplikace
- Osnova
- Návrh
- Data
- Ovladače
- XML editor
- Události
- Chyby
- Vlastnosti



*Obr. 2: Perspektiva pro vývoj XPages*

zdroj: vlastní printscreen Domino Designeru

Seznam aplikací je stále stejný jako při vývoji pro klienta. Pod ním lze vidět osnovu. Ta poskytuje přehled o hierarchii jednotlivých vložených prvků. Jednoduše se tak dají uchopit a přesouvat přesně tam kam je potřeba, nebo je mazat.

Dole jsou k nalezení tři důležité záložky. Vlastnosti, události a chyby. Skrze vlastnosti se nastavují vlastnosti vložených prvků nebo i celých Xpage. Podle toho, jaký je to prvek, se zobrazují určité vlastnosti, které lze nastavit. Obvykle je možné nastavit název prvku, styl, viditelnost, datový zdroj a další. Základní se dají nastavit pouhým zaškrtnutím, či výběrem z několika možností. Složitější hodnoty se musí nastavit jako vypočítávané. Asi nejdůležitější položka vlastností je: all properties. Zde se v přehledné tabulce zobrazí, všechny vlastnosti, které mohou být u prvku nastavené.

V událostech se programuje, co se má stát při určité akci. Například co se stane při kliknutí na prvek, po najetí myši. Jsou tu i události pro celé Xpage, jako co se má stát při otevření stránky, zavření stránky. Jsou obdobné jako události u formulářů (QueryClose, QueryOpen, PostOpen, PostClose apod.). Dále tato záložka umožňuje nastavení AJAX prvku částečná aktualizace. Lze díky tomu určit element, který se má aktualizovat a zbytek stránky zůstane stejný. Tím se snižuje množství přenesených dat mezi serverem a webovým prohlížečem.

Záložka chyby slouží k zobrazení chyb a varování, které zabraňují správnému fungování aplikace. Kontrola chyb běží stále a minimalizuje dobu řešení problémů. Je potřeba stále ji kontrolovat, protože když XPage nebo Custom Control obsahuje chybu, tak se úpravy neuloží.

V datech se zobrazují po nadefinování datového zdroje prvky datového zdroje, které mohou být pomocí „Drag and Drop“ vkládány. Pod ním je zobrazen seznam všech ovládacích prvků, které lze vložit.

### **2.2.2 Programovací jazyk**

Základním jazykem je JavaScript obohacený o @Funkce, což programátorům Lotus Notes ulehčuje práci. JavaScript je skriptovací jazyk, pomocí kterého lze vytvářet ovládací prvky uživatelského rozhraní, ověřování vstupů od uživatele, vlastní logiku aplikace. Obvykle se vkládá do HTML, ale v případě XPages do XML. JavaScript může být spouštěn jak na straně serveru, tak na straně klienta. Na straně klienta slouží například k ověření zadaných dat, aniž by se formulář musel odesílat na server a zpět klientovi, pokud zadaná data nejsou zcela správná. Serverový JavaScript slouží například k ukládání a mazání dokumentů. Příkladem použití obou JavaScriptů jsou události. Události mohou být jak na straně serveru, tak na straně klienta. Uveden zde bude krátký příklad pro událost kliknutí na tlačítko uložit. Pod tlačítko uložit se přidá akce pomocí serverového JavaScriptu pro uložení dokumentu a na straně klienta se zadá JavaScript pro ověřování zadání některých z hodnot. Při kliknutí na tlačítko se nejprve spustí klientský JavaScript a v případě nesplnění podmínky se proces ukládání zastaví. Pokud podmínka bude splněna, následuje serverový JavaScript, který odešle dokument na server a uloží jej v aplikaci.

## **2.3 Rozdíly při vývoji XPages a klientských aplikací Lotus Notes**

Oba způsoby vývoje jsou velmi rozdílné. V následujících kapitolách budou popsány nejzákladnější rozdíly.

### **2.3.1 Formulář**

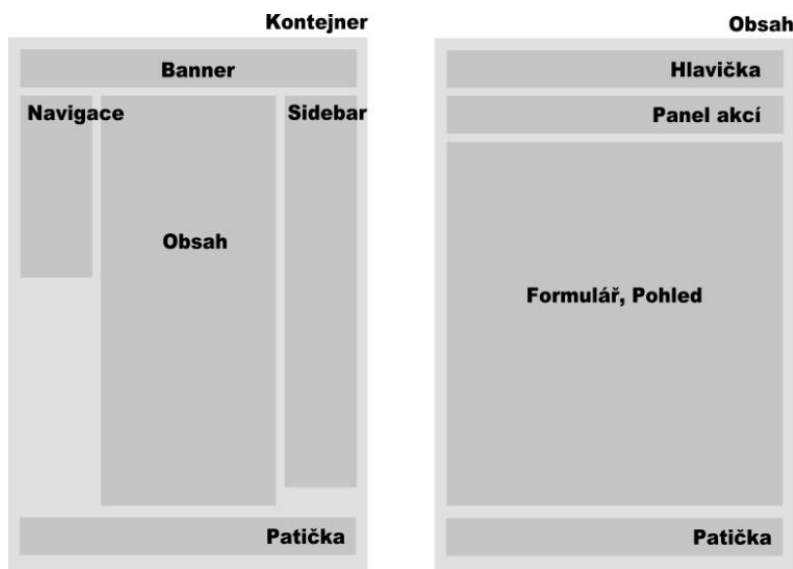
Rozdílem mezi klasickou klientskou aplikací a XPages aplikací je, jakým způsobem vnímají formuláře. V klasické aplikaci formulář určuje strukturu dat, grafické rozhraní a zároveň logiku aplikace. XPages aplikace ukládá grafické rozhraní a logiku v několika XPages a formuláře jsou odpovědné pouze za to, jak budou data strukturována a ukládána.

### 2.3.2 Rozložení prvků v aplikaci

Vzhled každé aplikace je určen rozložením prvků. Stanovuje se, kde bude navigace, panel akcí, samotný obsah a další. Obvykle má aplikace vymezený jednotný vzhled, který je pro přehlednost zachován v celé aplikaci. U aplikací určených pro klienta, se rozložení navrhuje pomocí rámců.

Rámce se vytvářejí v Domino Designeru pomocí vizuálních nástrojů. Při vytváření se zvolí počet rámců, a jak by měli být rozloženy. Tento výběr není limitující, rámce lze odebírat a přidávat v průběhu návrhu. Ke každému rámcu se zvolí, co má obsahovat. Může to být osnova, pohled, formulář, stránka i jiný rámec. Společně tyto rámce vytvoří jednu sadu rámců, která určuje výsledné rozložení aplikace. Poté se pouze zvolí ve vlastnostech aplikace, který rámec má být výchozí a při otevření aplikace se automaticky zobrazí. Jednotlivé formuláře obsahují vlastní rozložení, které se obvykle stanovuje pomocí tabulek. [4]

U XPages se rámce nepoužívají a rozložení se stanovuje jiným způsobem. Rámce vymezují jednotlivé panely. Jejich velikost a pozice je obvykle učena pomocí CSS, tak jak je potřeba, či podle toho co se komu líbí. Obsah jednotlivých panelů se nejčastěji vytváří v Custom Controls a poté se vkládají do panelů na XPage. Úvodní stránka obsahuje oblasti jako obsah, banner, navigaci, sidebar (postranní sloupec), patičku a to celé je ohraničeno kontejnerem. Obsah jednotlivých stránek stránky se dělí na oblasti jako hlavička, panel akcí a obsah. Nejčastější rozložení, které se používá, je nakresleno na *Obr. 3: Rozložení rámců*



Obr. 3: Rozložení rámců

zdroj: vlastní

### 2.3.3 Navigace

Navigace umožňuje uživatelům pohybovat se v aplikaci. Při vytváření klientské aplikace se využívá osnov. Pro vytvoření nové osnovy je k dispozici tlačítko, pomocí kterého program Domino Designer vygeneruje osnovu aplikace obsahující všechny pohledy. Jednotlivé položky s pohledy se mohou mazat, zakládat, přesouvat, vnořit či seskupovat do složek. Poté se může kliknout na tlačítko „použít osnovu“, díky kterému se okamžitě vytvoří nová stránka, obsahující osnovu. K dokončení již stačí stránku vložit do rozložení rámců.[4]

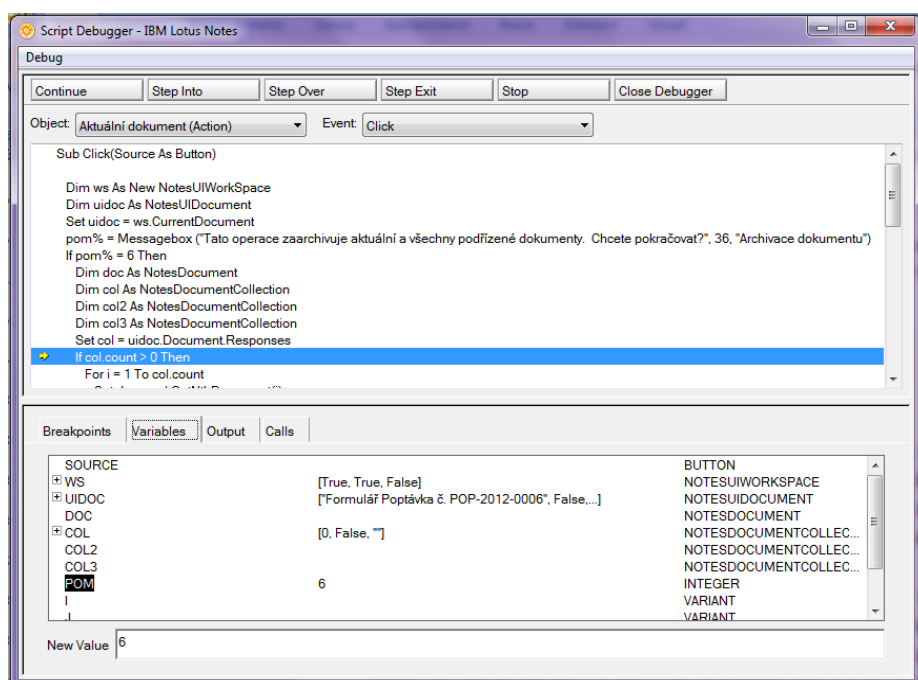
Při vytváření aplikace pomocí XPages, není k dispozici žádný prvek pro rychlé vytvoření. Osnova se vytváří pomocí linků, které otevrou požadovanou XPage, nebo Custom Control.

### 2.3.4 Ladění aplikace

Při programování je velice důležité, aby se aplikace dala ladit. Existuje pro to několik způsobů. Jejich použití se liší podle toho, v jakém jazyce programujeme.

U jazyka funkcí neexistuje nástroj pro ladění. K ladění lze použít příkaz @Prompt, který vypíše hodnotu požadované proměnné.

LotusScript má k dispozici debugger pro ladění. Pokud je potřeba ladění zapnout musí se v Lotus Notes zaškrtnout „Debug LotusScript“, pod volbou nástroje. Nyní se debugger spustí, vždy když začne běžet skript. Jak vypadá, můžete vidět na *Obr. 4: Debugger pro LotusScript*.



*Obr. 4: Debugger pro LotusScript*

zdroj: vlastní printscreen debuggeru

Uprostřed se nachází oblast s kódem, na které je zvýrazněno, kde se právě v kódu nacházíme. Lze zde umístit breakpointy, pro rychlejší ladění. Dole je několik záložek, z nichž nejdůležitější je variables. Tam se nacházejí informace o proměnných, zejména jakých nabývají typů a hodnot. Nahoře jsou tlačítka pro krokování. Pod nimi se zobrazuje označení prvku a události, které právě ladíme. Na *Obr. 4: Debugger pro LotusScript* je vidět, že se jedná o akci aktuální dokument a událost kliknutí.

U XPages není nástroj pro ladění. V průběhu vytváření aplikace je občas nutné zjistit hodnotu určité proměnné. Například, když je potřeba zkontrolovat hodnotu vzorce @Name('[CN]', @UserName()), zda vypisuje správně hodnotu autora. Využívá se pro to

vypočítávaných polí. Stačí požadovaný kód vložit do vypočítávaného pole a v něm se hodnota zobrazí.

### **2.3.5 Rozdílné přihlašování**

K přihlášení do klienta Lotus Notes je zapotřebí ID uživatel. ID uživatele je v podstatě licence, která dovoluje využívat Lotus Notes. Obsahuje informace, které jednoznačně identifikují uživatele v systému. Tato licence umožňuje přístup k poštovní schránce, kalendáři, osobním úkolům, kontaktům a dalším zakoupeným, či vyvinutým aplikacím.

Přehled o všech uživateliích a skupinách je v aplikaci Adresní kniha. Pro založení nového uživatele je třeba otevřít aplikaci skrze Domino administrátora a zde vytvořit ID s heslem. Pro přihlášení se musí ID dát na uživatelský počítač do složky data. Přihlášení je pak stejné jako do každé jiné aplikace. Při spuštění vás aplikace vyzve k zadání uživatelského jména a hesla, při správném zadání se klient Lotus Notes spustí.

Pro XPages aplikaci jsou podmínky stejné, každý uživatel se přihlásí, až po vytvoření licence. Při otevírání aplikace přes prohlížeč se zobrazí výchozí přihlašovací okno prohlížeče. Pomocí aplikace Domino Web Server Configuration lze vytvořit jednotnou přihlašovací stránku, kterou lze dále editovat. Aplikace se vytváří ze šablony, která je na Domino serveru a obsahuje formulář: \$\$LoginUserForm, sloužící jako přihlašovací stránka. Pro dokončení nastavení se aplikace otevře přes klienta Lotus Notes a vytvoří se dokument pro mapování. Uživatelské jméno je stejné jako pro klienta. Heslo je jiné a nastavuje se v adresní knize pod volbou Internet Password. Obsahuje-li aplikace odkaz na změnu hesla, může si jej uživatel upravit sám. V jiném případě má možnost změnit heslo pouze administrátor v adresní knize.

U XPages je ještě další možnost jak povolit přístup do aplikace i uživatelům bez ID. Nový uživatel se založí v externí adresní knize, kde se mu zadá přihlašovací jméno a heslo. Nesmí se zapomenout na zařazení těchto uživatelů do ACL, jelikož u každé aplikace je zakázán přístup pro uživatele, kteří v ní nejsou.



## **3 Převod stávajících aplikací a tvorba nových**

Způsob převodu a tvorba nových aplikací, pomocí XPages, se od sebe moc neliší. V podstatě i u převodu se musí vyvinout celá aplikace znovu. Neexistují žádné nástroje, které by převod zajistily. U převodu do XPages se obvykle vyvíjí pouze webové rozhraní a aplikace v klientovi zůstává stejná. Vzniká tím tak problém, když se aplikace upravuje. Musí se změnit návrh klientské aplikace i XPages. Výhodou u tvorby webového rozhraní je sdílení dat. Data v aplikaci jsou uložena ve formulářích a pohledech, a pokud se správně propojí s XPages, tak není potřeba je do webové aplikace nijak převádět, či znovu vytvářet. Při převádění aplikace je dobré dbát na to, aby byla co nejvíce podobná aplikaci vytvořené pro klienta Lotus Notes. Uživatelé tak v ní mohou pracovat stejně jako v té, na kterou už byli zvyklí a nemusí se učit téměř nic nového. Jen některé funkčnosti se musí upravit, aby v XPages správně fungovali.

### **3.1 Tvorba nové aplikace**

Při vytváření nové aplikace je výhoda, že ji lze vytvořit, aby se dala otevřít jak přes webový prohlížeč, tak skrze klienta. Nemusí se tak vytvářet návrh pro klientskou aplikaci a webové rozhraní zvlášť. Z toho vyplývá výhoda v podobě jednodušších úprav, které stačí provést pouze jednou v XPages. V kapitole 4 bude popsán vývoj ukázkové aplikace určené jak pro klienta, tak webový prohlížeč.

#### **3.1.1 Postup tvorby nové aplikace**

Při tvorbě nové aplikace se nejprve vytvářejí formuláře a pohledy, které budou stanovovat strukturu dat. Poté se může vytvořit úvodní stránka, nebo se vytvářejí jednotlivé XPages a Custom Controls, které budou data z formulářů a pohledů zobrazovat. U úvodní stránky se nejprve stanoví rozložení jednotlivých prvků, jako navigace, obsah, panel akcí apod. Poté se jednotlivé prvky rozložení vytvářejí, buď přímo na úvodní stránce, nebo pomocí Custom

Control/XPage, které se do úvodní stránky po dokončení vloží. Jako poslední se vytváří logika celé aplikace. Například co se má stát při kliknutí na tlačítko a další.

## **3.2 Převod stávajících aplikací**

U převodu stávajících aplikací je potřeba klást důraz na správné datové propojení s formulářem, aby se data již vytvořená zobrazovala ve webové aplikaci správně.

### **3.2.1 Postup převodu aplikace již vytvořené pro klienta**

Postup převodu stávajících aplikací je velice podobný jako tvorba nových. Pouze se již nemusí definovat formuláře a pohledy, jelikož jsou vytvořené. Začíná se obvykle s návrhem úvodní stránky a jejích jednotlivých prvků. Poté se postupně vytvářejí XPages nahrazující formuláře a Custom Controls zobrazující jednotlivé pohledy.

V rámci mé praxe se uskutečnil převod Obchodních aktivit (dále jen OA) do XPages. OA jsou poměrně rozsáhlá aplikace, proto se jako první fáze zvolila převést je do XPages pouze pro čtení. Poté až bude dokončena tato fáze, přidávat jednotlivé funkcionality do XPages, aby se z nich stala plnohodnotná webová aplikace. V následujících kapitolách bude naznačen konkrétní průběh převodu aplikace do formy pro čtení. Některé části jsou přiměřeně zkráceny a zobecněny, jelikož kdyby se měl podrobně napsat celý postup, bylo by to velice rozsáhlé. Popsán bude způsob převodu úvodní stránky, jednoho formuláře a jednoho pohledu.

Pro přehlednost je zde naznačen postup převodu OA:

- Založení úvodní stránky.
- Zvolení rozložení jednotlivých panelů na úvodní stránce.
- Poté se postupně vytvářejí jednotlivé části rozložení. Nejprve navigace a pak vlastní obsah stránky tvořený pohledem.
- Vytvoření XPage zobrazující obchodní případ.

- Propojení XPage pro obchodní případ s formulářem Obchodní případ.
- Dále se vytvářejí jednotlivé části obchodního případu:
  - panel akcí
  - hlavička
  - obsah - složený ze základních údajů, kontaktů, poznámek, dalších údajů a souborů, úkolů a vývojového stromu.
  - zápatí
- Založení Custom Controlu pro pohled.
- Propojení pohledu s pohledem vytvořeným pro klienta.
- Vytváření jednotlivých sloupců pohledu.

Tento postup by se dal opakovat pro každý formulář a pohled OA, dokud by nebyly všechny vytvořené.

### 3.3 Aplikace Obchodní aktivity

Aplikace OA slouží k sledování průběhu obchodních případů. Vkládají se sem emaily, záznamy z jednání, telefonátů, smlouvy atd. Vzniká tím podrobná dokumentace případu, která zabezpečuje případné bezproblémové předání obchodního případu jinému zaměstnanci. Zaznamenávají se zde i akvizice, jež nelze považovat za obchodní případ. Jedná se například o oslovení potencionálního zákazníka, nebo prezentaci u firmy. Za obchodní případ se považuje až v případě, že se jedná o konkrétním produktu, cenových podmínkách apod. Zapisují se zde jednotlivé fáze od poptávky přes nabídku, smlouvu až po zakázku. Pod jednotlivými fázemi se poté evidují přijaté a odeslané emaily, jednání, faxy, úkoly k případu. Je zde provázanost s poštovní schránkou a osobním kalendářem. Díky tomu lze přímo z aplikace odesílat e-maily, a v případě zapsání jednání se automaticky vytváří záznam v kalendáři.[15]

V této aplikaci jsou důležitá přístupová práva. Nejen přístup k celé aplikaci pomocí ACL, ale také přístup k jednotlivým obchodním případům. Tento přístup je nastaven pomocí polí ve formuláři typu čtenáři a autoři. Tyto pole mohou obsahovat jak jednotlivé uživatele, tak skupiny. Pokud budete chtít pole vytvořit, vkládají se do formuláře jako klasická pole s tím

rozdílem, že jejich typ se nastaví na authors (autoři), nebo readers (čtenáři). Poté už Lotus Notes bude vědět, že se jedná o pole, které slouží k nastavení přístupových práv. Uživatelé v poli autoři mohou dokument otevřít i upravovat. Čtenáři mohou dokument pouze otevřít pro čtení. Pokud by pole čtenáři zůstalo prázdné, dokument mohou číst všichni, kdo mají přístup k aplikaci, když ale stanovíme čtenáře, jen oni a autoři mohou dokument číst.

### 3.3.1 Úvodní stránka

Jako první se vytvořila úvodní stránka, která se zobrazuje při spuštění aplikace. Je tvořena pomocí XPage, která se nejprve musí založit.

Existují dva způsoby jak vytvořit novou XPage. První jde provést skrze seznam aplikací, kde pravým tlačítkem lze kliknout na XPages a zvolit „New XPage“. Druhou možností je dvojklikem kliknout na XPages, tím se otevře okno se seznamem již vytvořených XPages a nahoře je tlačítko pro založení nové XPage. Obdobně se vytvářejí také Custom Controls.

Při vytváření je třeba zadat jméno, které může obsahovat písmena, čísla nebo podtržítka. Lze přidat komentář, který blíže specifikuje danou stránku. V této chvíli je možno rovnou nadefinovat datový zdroj, zaškrtnutím možnosti: „Add data source to the page“. Není to ale povinné, datový zdroj lze nadefinovat kdykoli při návrhu XPage. Teď již stačí stisknout „OK“ a nová Xpage je vytvořena. Vytvořená XPage je prázdná. Pouze při pohledu na záložku source, je tam již vydefinován kód v XML (zobrazen na *Obr. 5: Source*). XML se automaticky generuje v průběhu vytváření XPage.



Obr. 5: Source

zdroj: vlastní printscreen aplikace Domino Designer

Nyní se začne s návrhem rozložení úvodní stránky. Rozložení je vytvořeno podle schématu, o kterém se píše v kapitole 2.3.2. Pouze jsou vynechány části jako patička a sidebar. Sidebar tam není z důvodu, že není potřeba a také je lepší využít tento prostor k rozšíření obsahu stránky.

### 3.3.2 Rozložení Úvodní stránky

Celé rozložení je tvořeno panely. Umístění jednotlivých panelů se řídí kaskádovými styly. Ty je nejprve potřeba vytvořit a vložit do zdrojů úvodní stránky. CSS se zakládají v nabídce zdroje, list se styly a zde založit nový. Vznikne tak textový editor, do něhož lze vkládat CSS. Je tu i barevné rozlišení kódu pro větší přehlednost. Hotový soubor s CSS se do XPage vloží skrze záložku style, pomocí tlačítka „Add style sheet to page“. Nyní lze jednotlivým prvkům přiřazovat kaskádové styly. Ukázka části použitých kaskádových stylů:

```
h1{ /*použito pro nadpis v banneru*/
    background:url(bannerBackground.gif) transparent;
    border: 2px groove #4682b4;
    font-size:14pt;
    padding:7px 0 7px 10px;
}
div.banner {
    padding:8px 8px 16px;
    height: 60px;
}
div.navigation { /*levé menu*/
```

```

background:#ffffff url(./Outline.jpg) top left repeat-x;
border:2px groove #4682b4;
float: left;
margin: 0 8px;
padding: 8px;
width: 150px;
}

```

*Ukázka kódu 3: Kaskádové styly*

Některé vlastnosti v CSS nadefinované nejsou. V Domino Designeru se základní vlastnosti jako odsazení, styl písma, pozadí dají nastavit přímo v návrhu. Slouží k tomu záložka style. V ní lze jednoduše pomocí grafického rozhraní nastavit požadované hodnoty, na jejichž základě se vytvoří odpovídající XML značky. Například na *Ukázka kódu 4: XML kód pro pozadí v banneru* je vidět XML banneru (slouží k zobrazení názvu celé aplikace), u kterého se pomocí záložky style nastavila barva pozadí. Automaticky se tak vydefinoval druhý řádek se stylem.

```

<xp:panel id="banner" styleClass="banner"
    style="background-color:rgb(114,180,207)" ">
</xp:panel>

```

*Ukázka kódu 4: XML kód pro pozadí v banneru*

### 3.3.3 Navigace

V levé části aplikace se nachází navigace. Obsahuje hodně pohledů, které jsou rozděleny do jednotlivých složek pro větší přehlednost. Složky se vytvářejí pomocí kontejnerového prvku sekce. V sekcích se nacházejí panely obsahující linky pro každý řádek navigace. Linky obvykle slouží k přesměrování na jinou stránku. Zde jsou ale využity pro uložení názvu Custom Controlu, ve kterém je vložen požadovaný pohled, jenž se má zobrazit. Tato hodnota se pak předá prvku include page, vloženém v obsahu indexu. Include page slouží k vložení XPage i Custom Controls do jiné XPage. Díky tomu se při pohybu v aplikaci pomocí navigace nepřepočítává celá stránka, ale pouze prvek include page.

```

sessionScope.put("ccToBeDisplayed", "ccViewOPDleKratNazvu.xsp");
context.redirectToPage("index.xsp");

```

*Ukázka kódu 5: JavaScript pro link*

Konkrétně kód na *Ukázka kódu 5: JavaScript pro link* zajistí, aby se stránce index předal *ccToBeDisplayed* s hodnotou Custom Controlu *ccViewOPDleKratNazvu*. V indexu je vložen prvek *include page*, který se dopočítá podle hodnoty *ccToBeDisplayed*, uložené v *sessionScope*. *SessionScope* nám umožňuje sdílet hodnoty napříč stránkami po dobu trvání *session*. *Session* například zanikne při odhlášení, ukončení aplikace.

```
if ( !sessionScope.get("ccToBeDisplayed") ) {  
    "ccViewOPDleFirem.xsp";  
} else {  
    sessionScope.get("ccToBeDisplayed");  
}
```

*Ukázka kódu 6: JavaScript pro include page*

V JavaScriptu na *Ukázka kódu 6: JavaScript pro include page* je vidět, že výchozí pohled je *ccViewOPDleFirem*, protože se zobrazí vždy, pokud je hodnota *ccToBeDisplayed* prázdná.

Výsledný vzhled indexu je zobrazen v příloze A.

### 3.3.4 Obchodní případ

Obchodní případ slouží k zaznamenání průběhu jednotlivých fází obchodního případu. Je to formulář, který se bude postupně vytvářet pomocí *XPages*, aby se dal přecházet přes webový prohlížeč.

První krok je vytvoření *XPage* pro obchodní případ. Dále se nesmí zapomenout na nastavení, které se musí provést téměř vždy při vytváření jakékoliv *XPage*. Je to nastavení, aby se v prohlížeči místo formuláře zobrazovala odpovídající *XPage*. Slouží pro to hodnota ve vlastnostech formuláře: *On web access*, kde se vybere příslušná *XPage*. Toto nastavení formuláře je vidět na *Obr. 6: Zobrazení XPage místo formuláře*

*Obr. 6: Zobrazení XPage místo formuláře*

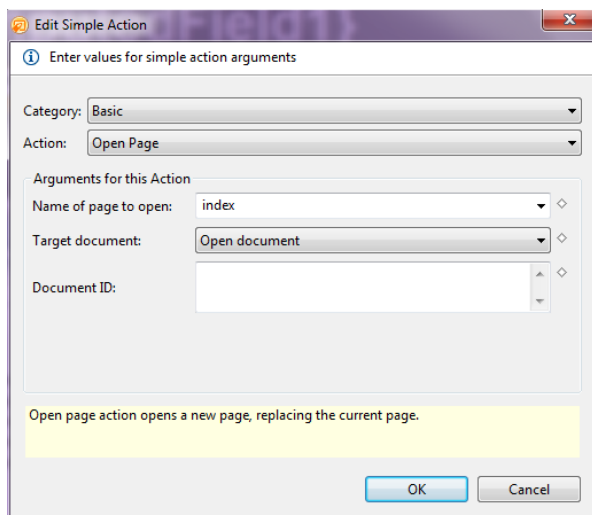
zdroj: vlastní printscreen vlastností formuláře

Rozložení obchodního případu se skládá z panelu akcí, hlavičky, obsahu, a záhlaví. Toto rozložení je vytvořeno opět pomocí panelů, jejichž velikost a umístění je určeno kaskádovými styly. Výsledná XPage s obchodním případem je zobrazena v příloze A.

### 3.3.5 Panel akcí

Panel akcí obvykle obsahuje tlačítka pro uložení, editaci a zavření. V obchodním případě je vloženo pouze tlačítko zavřít. Aby tlačítko správně fungovalo, musí se nastavit událost při kliknutí na objekt. Tato událost je jednoduchá akce, která se nastaví skrze okno zobrazené na *Obr. 7: Nastavení jednoduché akce*. Akce nám otevře XPage index, který nahradí stránku s obchodním případem. V jednoduchých akcích se dají také vytvořit další akce jako uložení dokumentu, smazání dokumentu, spuštění skriptu.





*Obr. 7: Nastavení jednoduché akce*

zdroj: vlastní printscreen okna pro nastavení jednoduchých akcí

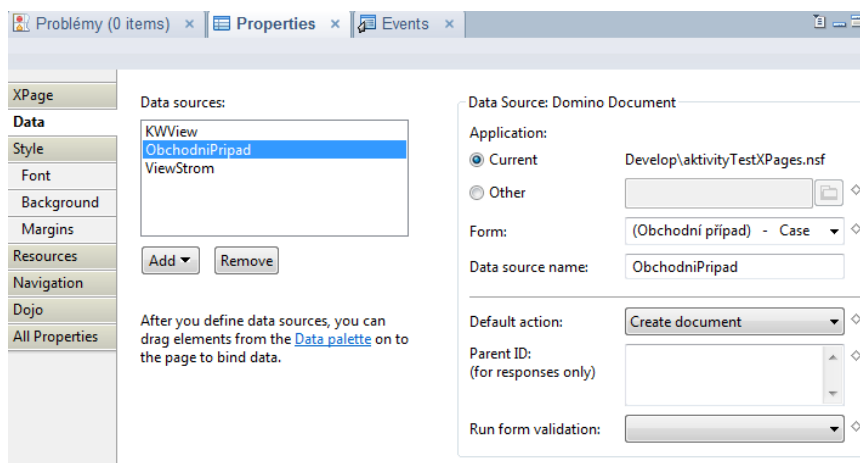
### 3.3.6 Hlavička

Je vytvořena pomocí Custom Controlu. Skládá se z tabulky, která obsahuje nejdůležitější data z obchodního případu. Například název případu, odpovědnou osobu za celý případ, číslo zakázky.

### 3.3.7 Obsah obchodního případu

Většinu stránky obchodního případu tvoří vlastní obsah. Pro lepší přehlednost jsou jednotlivé informace rozděleny na karty. Jednotlivé karty se jmenují: základní údaje, kontakt, poznámky, další údaje a soubory, úkoly a vývojový strom. Jsou vytvořeny pomocí kontejnerového prvku tabbed panel.

Základní údaje obsahují informace jako název případu, cena, úspěšnost, termín realizace. Data jsou uspořádána v tabulce, kde první sloupec je tvořen popisky a druhý poli zobrazující hodnoty. Pole na formuláři i v XPage musí zobrazovat stejné hodnoty. Proto je nutné nadefinovat pro XPage datový zdroj, který tak oba prvky propojí. Pokud se datový zdroj nenastavil již při zakládání XPage, může se nadefinovat na záložce data (zobrazené na *Obr. 8: Datový zdroj XPage*) ve vlastnostech celé XPage.



Obr. 8: Datový zdroj XPage

zdroj: vlastní printscreen vlastností XPage

Tady se vybere jako datový zdroj konkrétní formulář a nazve se. Tento název se pak může používat jako proměnná v JavaScriptu. Při správném nadefinování zdroje lze každému vytvořenému poli na XPage přiřadit odpovídající hodnotu pole na formuláři. Některá pole se již při propojení zobrazí stejně jako na formuláři, ale ostatní se musí dále upravit. Týká se to zejména polí, které jsou na formuláři vypočítávané pomocí @Funkcí, nebo jsou vytvořené složením více polí. Tyto hodnoty se také na XPage musí vypočítat, ale pomocí JavaScriptu.

Například je tomu tak u pole s cenou. To se skládá ze dvou hodnot: cena a měna. Pomocí JavaScriptu, zobrazeném na *Ukázka kódu 7: JavaScript pro pole cena*, se tak spojili obě pole do jednoho a ještě se ošetřil případ, když hodnota bude prázdná, aby se zobrazovalo 0.00, stejně jako v aplikaci pro klienta.

```
var cena = ObchodniPripad.getItemValueInteger("Cena")
var mena = ObchodniPripad.getItemValueString("Mena")
if (cena == 0){
return "0.00"
} else if (mena == ""){
return cena
} else {
return cena + " " + mena
}
```

Ukázka kódu 7: JavaScript pro pole cena

Následující karta zobrazuje kontaktní údaje o firmě, ke které se konkrétní případ váže. Vytvořena je dalším tabbed panelem se třemi kartami. V každé kartě je vložen Custom

Control, jelikož jsou použity na více XPages. Vzhled celé karty je vidět na *Obr. 9: Kontakty*.

*Obr. 9: Kontakty*

zdroj: vlastní printscreen aplikace Obchodní aktivity

Všechny tabulky na kartě kontakty jsou tvořeny pomocí vloženého Custom Controlu. U Custom Controls se datové propojení definuje jinak, jelikož se mohou vyskytovat na více místech. Jednotlivé hodnoty se vypočítávají pomocí vzorce: `@GetField("fieldName")`, kde *fieldName* je jméno pole. Zdroj tak není spojen s konkrétním formulářem, ale s polem, které se dopočítá podle zdroje XPage, na které je Custom Control vložen.

Takto jsou datová propojení vytvořena na všech polích Custom Controls. Opět u některých polí nestačí jednoduché propojení pomocí funkce `@GetField` a hodnota musí být dopočítána pomocí JavaScriptu. Například pole pro zobrazení firmy je vypočítáváno z několika polí na formuláři pomocí následujícího jazyka funkcí:

```
@If(Forma = ""; JmenoFirmy; JmenoFirmy + Oddelovace + Forma)
```

*Ukázka kódu 8: Jazyk funkcí pro pole firma*

*Ukázka kódu 8: Jazyk funkcí pro pole firma* slouží k zobrazení názvu firmy, který se skládá ze jména firmy, oddělovače a právní formy podniku. Je tam podmínka, že pokud u firmy není zadána její právní forma, zobrazí se pouze její jméno. Tato formule se tak musí přepsat do odpovídajícího JavaScriptu (zobrazený je na *Ukázka kódu 9: JavaScript pro pole firma*), aby se pole správně zobrazovalo.

```

var operators = ObchodniPripad.getItemValueString("Oddelovace");
var jmeno = ObchodniPripad.getItemValueString("JmenoFirmy");
var forma = ObchodniPripad.getItemValueString("Forma")

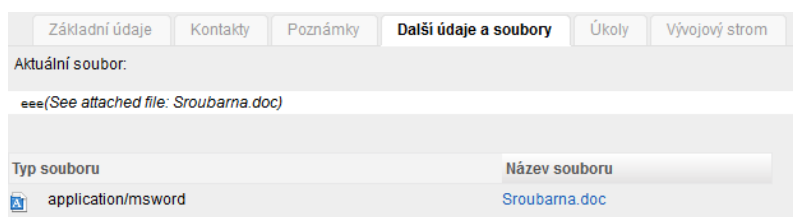
if (forma == ""){
    firma = @Text(jmeno)
    return firma;
} else {
    firma = @Text(jmeno) + @Text(operators)+ @Text(forma)
    return firma;
}

```

*Ukázka kódu 9: JavaScript pro pole firma*

Karta poznámky slouží k zobrazení poznámek. Poznámek může být několik od různých uživatelů. Ve formuláři pro klientskou aplikaci jsou poznámky vytvořeny pomocí pole, u něhož je ve vlastnostech povolena volba zobrazovat vícenásobné hodnoty, a jako oddělovač se nastavil nový řádek. V XPage se pro to používá kontejnerový prvek repeat. Repeat umožňuje zobrazovat ve sloupcích, nebo řádcích několikanásobné hodnoty.

Následuje karta další údaje a soubory. Je tvořena dvěma rich text poli, které umožňují zobrazovat formátovaný text a soubory. Pro správné zobrazení vložených souborů v XPages se používá prvek file download (na *Obr. 10: File download*). Do něj se stáhnou všechny soubory z pole, na které je propojený. Přílohy se dají odtud i otevřít či uložit na disk.



*Obr. 10: File download*

zdroj: vlastní printscreen prvku file download

Předposlední karta úkoly obsahuje odkazy na úkoly vytvořené k obchodnímu případu. Tento odkaz v aplikaci pro klienta vás přepojí do aplikace Úkoly, kde se dá úkol prohlédnout. Díky tomu, že aplikace Úkoly je vytvořená pouze pro klienta, ve webovém prohlížeči tento odkaz ztrácí smysl. Pole s úkoly je tak vytvořeno jako rich text field, takže odkaz uvidíme, ale nelze ho otevřít.

Poslední záložkou je vývojový strom obchodního případu, lze se pomocí něj mezi jednotlivými částmi obchodního případu přepínat. Vývojový strom je vložený pohled, který má jako datový zdroj pohled obsahující všechny obchodní případy. Aby se v něm zobrazoval pouze konkrétní obchodní případ, musí se hodnoty vyfiltrovat pomocí unikátního čísla přiřazeného k obchodnímu případu. Lze tak v pohledu najít konkrétní obchodní případ. Kód pro filtraci je následující:

```
ObchodniPripad.getItemValueString("OPUniqueDocNum")
```

*Ukázka kódu 10: Kód pro filtraci pohledu*

### **3.3.8 Zápatí obchodního případu**

Zápatí formuláře obchodního případu je vytvořeno pro nastavení přístupových práv k dokumentu. Uživatel tam může zvolit stupeň utajení, a zda je dokument důvěrný. Při nastavování těchto položek se automaticky vytvářejí záznamy do skrytých polí authors a readers. Jako výchozí je zvolen všeobecný stupeň utajení. To znamená, že pole readers je prázdné a dokument mohou otevřít všichni, kdo mají přístup k aplikaci. V poli authors bude pouze autor dokumentu, který tak jako jediný může dokument i upravovat. Při nastavení, že dokument je důvěrný se do pole readers napíše autor dokumentu a ten jediný by mohl dokument číst a proto se tu zobrazí nabídka pro přidání dalších osob, které budou moci dokument otevřít. Aby toto utajení správně fungovalo i v XPages nemusí se nic nastavovat. Stačí, že nastavení je na formulářích vytvořených pro klienta a XPages toto nastavení automaticky přebírá.

### **3.3.9 Pohledy**

Pohledy slouží stejně jako v klientské aplikaci pro řádkové zobrazení všech dokumentů. Všechny pohledy jsou vloženy a vytvořeny jako samostatný Custom Control. V něm je vložen kontejnerový prvek pohled, datově propojený s odpovídajícím pohledem vytvořeným pro klienta. Jednotlivé sloupce se při datovém propojení zobrazují správně, pouze pokud jejich hodnoty nejsou vypočítávané. V opačném případě se musí hodnota

sloupce také vypočítat, ale pomocí JavaScriptu. Jako ukázkový zde bude uveden pohled dle firem, u kterého bude vysvětleno, jak jsou vytvořeny jednotlivé sloupce.

Při vkládání kontejnerového prvku pohled se nabízí možnost nadefinovat zdroj, který se nastaví na pohled: Obchodní aktivity/dle firem. Po potvrzení se zobrazí nabídka všech sloupců, jež pohled obsahuje a vybere se mezi nimi, které se mají zobrazit na Custom Control. Vyberou se sloupce zobrazující firmu, číslo zakázky, ikony, obchodní případy a odpovědného.

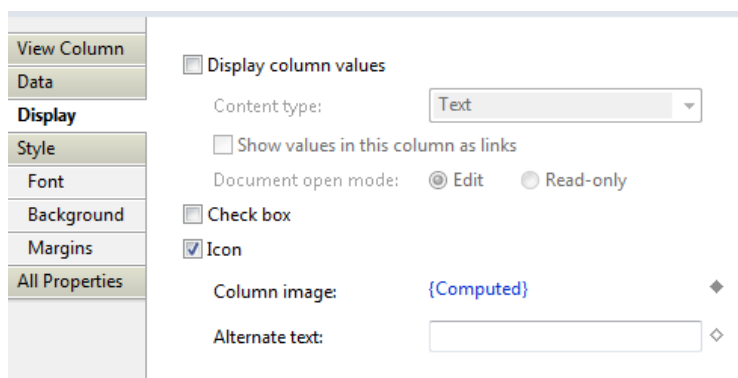
Sloupec pro zobrazení firmy je kategorizovaný. Znamená to, že každá firma je zobrazena jako jedna kategorie, pod kterou jsou všechny obchodní případy spojené s firmou. Kategorizovaný sloupec se vytváří v pohledu pro klienta, nastavením vlastnosti sloupce na categorized. Poté i sloupec v pohledu na Custom Controlu bude kategorizovaný.

V druhém sloupci jsou ikony k jednotlivým dokumentům pro jejich snazší hledání. Například pro přijatou korespondenci obálku. V pohledu pro klienta je tento sloupec tvořen čísly, která se vypočítávají na základě druhu dokumentu. Tyto čísla odpovídají názvům ikon, na které se pomocí jazyka funkcí přepočítávají. Toto se zajistilo pomocí JavaScriptu na *Ukázka kódu 11:Vypočítávané ikony*. Kde *view1* je název datového zdroje, ten obsahuje sloupec *ikony*, jenž zobrazuje čísla ikon. V případě, že hodnota sloupce je 0, tak se nemá zobrazit žádná ikona. V ostatních případech se číslo ve sloupci přidá za *vwicn* a vznikne název ikony. Názvy ikon jsou tedy například *vwicn001*, *vwicn002*, *vwicn003* atd. Tyto ikony jsou v adresáři, ke kterému je cesta uvedena na čtvrtém řádku kódu.

```
if (@Text(view1.getColumnValue("ikony"))=="0"){  
    return null;  
} else {  
    "/.ibmxspres/domino/icons/vwicn" +  
    @Right("00+@Text(view1.getColumnValue("ikony")),3) + ".gif"  
}
```

*Ukázka kódu 11:Vypočítávané ikony*

Tento JavaScript je vložen do vlastností sloupce na záložku display (zobrazené na *Obr. 11: Zobrazení ikon ve sloupci*). Ta určuje, jak se mají hodnoty ve sloupci zobrazit. Pro zobrazení ikon je zaškrtnuta volba „Icon“ a jejich hodnota je vypočítávaná.



Obr. 11: Zobrazení ikon ve sloupci

zdroj: vlastní printscreen vlastností sloupce

Další sloupec tvoří čísla obchodních zakázek, jelikož se zobrazoval správně, nemusí se dále upravovat. V následujícím sloupci jsou vidět jednotlivé obchodní případy. K obchodnímu případu patří další dokumenty jako nabídka, smlouva, email. Vytvářejí se z obchodního případu jako odpovědi na něj a podle toho se strukturují pod obchodním případem v pohledu. Aby tuto strukturu vytvářel pohled webové aplikace, musí se nastavit vlastnost ident response. Nachází pod vlastnostmi sloupce, a pokud je její hodnota true, sloupec identifikuje odpovědi a podle toho se zobrazí ve sloupci.

Poslední sloupec znázorňuje odpovědného. Zobrazoval se téměř správně, pouze ho bylo třeba doplnit JavaScriptem, aby se zobrazoval pouze pro obchodní případ a nezobrazoval pro podřízené dokumenty. Vypočítává se to na základě pomocného sloupce, který u obchodního případu má hodnotu 1.

```
if (@Text(view1.getColumnValue("PomocnySloupec"))=="1") {
    return (@Text(view1.getColumnValue("Odpovedny")));
} else {
    return null;
}
```

Ukázka kódu 12: Vypočítávaný sloupec s odpovědným

Dále se do všech sloupců dodělalo, aby se při kliknutí na ně otevřel požadovaný dokument. Toto nastavení se provádí skrze záložku display, která je vidět na Obr. 11: Zobrazení ikon ve sloupci. Tam se může povolit hodnota „Show values in this column as link“, která z hodnot sloupců udělá linky na dokumenty.

Výsledný pohled je vidět v příloze A. Ostatní pohledy se vytvářely obdobným způsobem pouze se změnou datového zdroje a některých sloupců.

## 4 Ukázka vývoje aplikace

V této kapitole bude popsán vývoj nové aplikace pouze v XPages. Aplikace bude určena pro web i klienta. Na rozdíl od aplikace OA, u které se pouze vytváří webové rozhraní a aplikace v klientovi zůstává naprosto stejná.

Pro ukázkou vývoje aplikace se vytvořila drobná aplikace Profily. Slouží k vytváření a zobrazování údajů o uživateli. Bude ukázáno, jak aplikace vytvořená pouze v XPages využívá klasických elementů návrhu, kterými jsou formuláře a pohledy, a také nových technologií jako Dojo a AJAX.

### 4.1 Vytvoření úvodní stránky

Index je úvodní stránka, která obsahuje navigaci aplikace, pohled s daty, tlačítka pro mazání a vytváření jednotlivých profilů. Rozložení prvků je stanoveno stejně jako v aplikaci OA. Nahoře je název aplikace. V levé straně je navigace, která je vytvořena obdobně jako u OA, a uprostřed se nachází samotný obsah obsahující pohledy. Výsledný vzhled úvodní stránky je v příloze B.

### 4.2 Profil

XPage s profilem obsahuje pole pro vyplnění údajů. Než se začnou jednotlivá pole vytvářet, musí se vytvořit formulář Profil, ve kterém budou vytvořena pole, která mají být na XPage. Jednotlivá pole se na formulář vloží bez jakéhokoliv formátování. Pouze se určí typ pole a jméno.

Pole na formuláři:

- Jméno
- Příjmení
- Email
- Telefonní číslo (typ: číslo)



- Datum narození (typ: Datum)
- Město
- Doplnující údaje (rich text pole)

Nyní se pole mohou vložit na XPage. Je proto potřeba u XPage nadefinovat datový zdroj, kterým bude právě formulář Profil. Po úspěšném nadefinování se pole z palety s daty pomocí „Drag and Drop“ umisťují do obsahu XPage. Vytvoří se automaticky tabulka s požadovanými poli a ke každému se vytvoří popisek s jeho názvem. Tabulka se přizpůsobuje obsahu, tak se upravil její vzhled, aby lépe využívala prostor stránky. Takže se šířka celé tabulky pomocí grafického rozhraní v Domino Designeru stanovila na 100%. Automaticky se tak vygeneroval příslušný kód pro vzhled v XML.

```
<xp:table style="width:100%">
```

*Ukázka kódu 13: XML pro styl tabulky*

Šířku a podobu jednotlivých sloupců s poli a popisky definují kaskádové styly.

```
.label {width:150px;height:25px /*popisky*/
}
.field {
    width:700px; /*pole*/
    padding-left:2.0px;
    font-family:sans-serif;
    font-size:9pt;
    color:rgb(0,0,0)
}
```

*Ukázka kódu 14: Ukázka použitých CSS*

Nad tabulkou pro zadání údajů je vytvořen panel akcí a nadpis stránky. V nadpisu je vypočítávané pole, jehož hodnota je dopočítána pomocí JavaScriptu (zobrazen na *Ukázka kódu 15: JavaScript pro vypočítávané pole*). V případě nového profilu zobrazí hodnotu: Nový profil, a v případě otevření již vytvořeného: „Profil“ + jméno.

```
profilName = getComponent("Jmeno").getValue();
profilPrijmeni = getComponent("Prijmeni").getValue()
if ( profilName == null ){
    return "Nový profil";
} else {
    return "Profil: " + profilName + " " + profilPrijmeni ;
}
```

*Ukázka kódu 15: JavaScript pro vypočítávané pole*

V panelu akcí jsou tlačítka pro zavření, uložení a editaci dokumentu. Tyto tlačítka mají vytvořené jednoduché akce k události při kliknutí na ně. Všechny tyto akce se odehrávají na straně serveru. Nastavují se pomocí okna na *Obr. 7: Nastavení jednoduché akce*. Pro tlačítko zavřít se zvolí akce pro otevření úvodní stránky. Dále musí nastavit jeho typ na cancel, aby aplikace věděla, že se nemá ověřovat správnost dokumentu, ale pouze ho zavřít. Tlačítko editovat má akci nastavenou na přechod do módu editace. U tohoto tlačítka je dále vhodné nastavit jeho viditelnost, aby se zobrazoval, když dokument není ve stavu editace. Pro skrývání všech prvků se používá vlastnost visible, která může být nastavena na true (viditelná vždy), false (vždy neviditelná), nebo je vypočítávána pomocí skriptu. Poslední tlačítko je pro uložení dokumentu. U něj se také nastavuje viditelnost, ale opačně než u tlačítka editovat. Tlačítko uložit je viditelné pouze v případě, že dokument je ve stavu editace. Pro uložení je dále vložena akce pro uložení. Zároveň se k této akci zadává parametr, jaká stránka se má po uložení otevřít, v tomto případě je to úvodní stránka.

```
!document1.isEditable();
```

*Ukázka kódu 16: Viditelnost tlačítka editovat*

```
document1.isEditable();
```

*Ukázka kódu 17: Viditelnost tlačítka uložit*

### 4.2.1 Ověřování zadaných hodnot

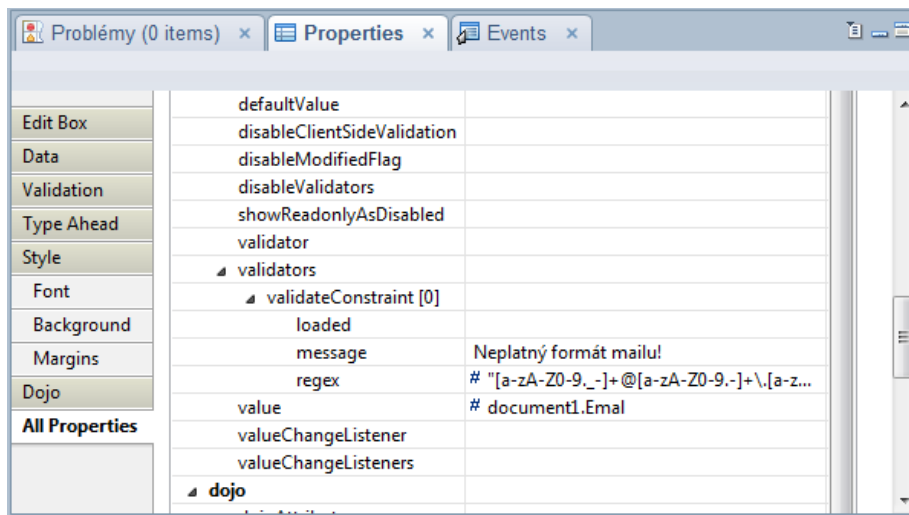
XPages mohou mít nadefinovány povinné a nepovinné položky, které jsou při ukládání kontrolovány. Není-li povinná hodnota vyplněna, XPage se neuloží a vypíše se chybové hlášení. Právě pole se jménem je povinné. Tato povinnost se nastavuje na záložce validation pod vlastnostmi pole. Zaškrtně se, že pole je povinné a připiše se požadovaná hláška, která upozorní, proč se dokument neuložil.

U polí se dále může kontrolovat jejich délka a syntaxe. Email má své náležitosti, proto je dobré ověřit, zda jeho syntaxe je správná. Ověření správnosti emailu se vytváří pomocí následujícího regulérního výrazu:

```
"[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}"
```

*Ukázka kódu 18: Regulérní Výraz pro email*

Tento výraz se nyní musí zakomponovat k poli email. K tomu se využívá validátorů. Ke každému poli návrhu může být přiřazen jeden nebo více validátorů, který kontroluje, zda zadaná hodnota splňuje stanovená kritéria. Tato validace se spustí při ukládání dokumentu, a pokud hodnota není správná, operace ukládání bude přerušena. Pro regulérní výraz se používá validátor typu validateConstraint. Do něj se vloží požadovaný regulérní výraz a zpráva, která se má zobrazit při nesplnění podmínky.



Obr. 12: Validace emailu

zdroj: vlastní printscreen vlastností pole

Všechny tyto validace se použijí na straně klienta. V případě, že by bylo nutné toto ověřování provádět na straně serveru, musela by se u každého pole nastavit vlastnost `disableClientSideValidation` (vidět je na Obr. 12: Validace emailu) na hodnotu `false`. XPage by se tak při ukládání odesílala na server, kde by se hodnoty kontrolovaly.

#### 4.2.2 Dojo

Jak u webových, tak klientských aplikací se lze setkat s malým kalendářem, který se zobrazí u pole pro výběr data. Díky tomu, že XPages využívají Dojo se tento kalendář vytváří velice snadno. Stačí hodnotu pole nastavit na datum a automaticky se k němu přidá tento kalendář. Tento prvek zjednodušuje uživatelům používání aplikace. Pomocí Dojo je také vytvořeno rich text pole pro zobrazení dalších údajů. V něm lze například formátovat text, vkládat obrázky.

### 4.2.3 Type Ahead

Další prvek pro zpříjemnění uživatelského rozhraní je vytvořen u pole pro výběr města. Je k němu díky podpoře AJAX vytvořen našeptávač obsahující města. Znamená to, že pokud se píše určitá hodnota pole, pod polem se zobrazuje seznam návrhů jako typ pro uživatele.



Obr. 13: Type Ahead pro pole město

zdroj: vlastní printscreen aplikace Profily

K povolení type ahead se přidává typeAhead tag k upravitelnému poli. Tento tag lze přidat přímo do XML nebo se jednodušeji povolí pomocí záložky type ahead, kde se zaškrtně volba „Enable Type Ahead“. Seznam typů pro pole může být přímo vepsán do pole návrhů nebo může být vypočítávaný. Častěji se objevuje vypočítávaný seznam, který se vytváří například podle sloupce pohledu. Díky tomu se hodnoty seznamu průběžně mění podle toho, co již do pole bylo zadáno a uloženo.

V případě měst je seznam stále stejný a je vytvořen na pomocném formuláři. Do návrhů se tedy přidá kód s cestou k poli na formuláři, ve kterém jsou hodnoty vloženy.

```
@DbLookup (@DbName (), "keywordLookup", "Mesta", "Values")
```

Ukázka kódu 19: Cesta k poli s městy

## 4.3 Pohledy

Pro dokončení aplikace zbývá ještě vytvořit Custom Controls s pohledy. Aplikace obsahuje pohledy:

- Všechny profily
- Profily dle jména
- Profily dle bydliště

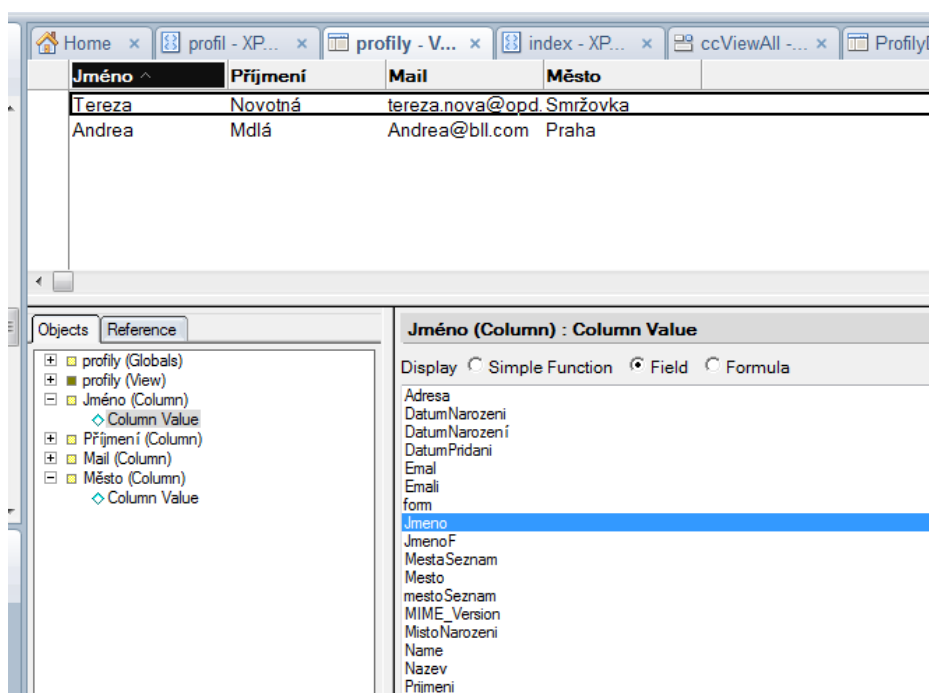
- Profily dle data narození

Pohledy se nejprve musí vytvořit stejně jako pro klientskou aplikaci. Postup pro vytvoření pohledu je následující. Nejprve založíme nový pohled. Do výběru pro pohled se napíše:

```
SELECT Form = "profil"
```

*Ukázka kódu 20: Výběr dokumentů pro pohled*

Tak se nastaví, že v pohledu budou zobrazovány formuláře profily. Teď už se jen v pohledu vytvoří sloupce, které se nazvou podle toho, jaké hodnoty zobrazují, a jejich hodnota se nastaví na pole ve formuláři. Vytvořený pohled v Domino Designeru je na *Obr. 14: Pohled v Domino Designeru*.



*Obr. 14: Pohled v Domino Designeru*

zdroj: vlastní printscreen aplikace Domino Designer

Tímto způsobem se vytvoří všechny pohledy, které se v aplikaci mají zobrazit. Pouze se vkládají jiné sloupce podle typu pohledu.

Dále se pro všechny pohledy založí jednotlivé Custom Controls. Jako datový zdroj se pro ně stanoví odpovídající pohledy. Nyní se na paletě s daty objeví všechny sloupce pohledu.

Ty lze pomocí „Drag and Drop“ vložit na Custom Control a automaticky se vytvoří pohled.

Ke každému pohledu se ještě vytvořili tlačítka pro mazání profilů a vytváření nových. Pro vytváření nového se vytvořila jednoduchá akce pro událost na kliknutí, která otevře nový prázdný profil. Tlačítko pro mazání je také vytvořeno jednoduchou akcí na kliknutí. Tato akce je nastavena na hodnotu „Delete Selected Documents“, a zajistí smazání vybraných dokumentů. K mazání lze přidat i potvrzující hlášku, aby dokumenty nemohli být smazány omylem.

Výsledná aplikace je zobrazena v příloze B.

## 5 Zhodnocení

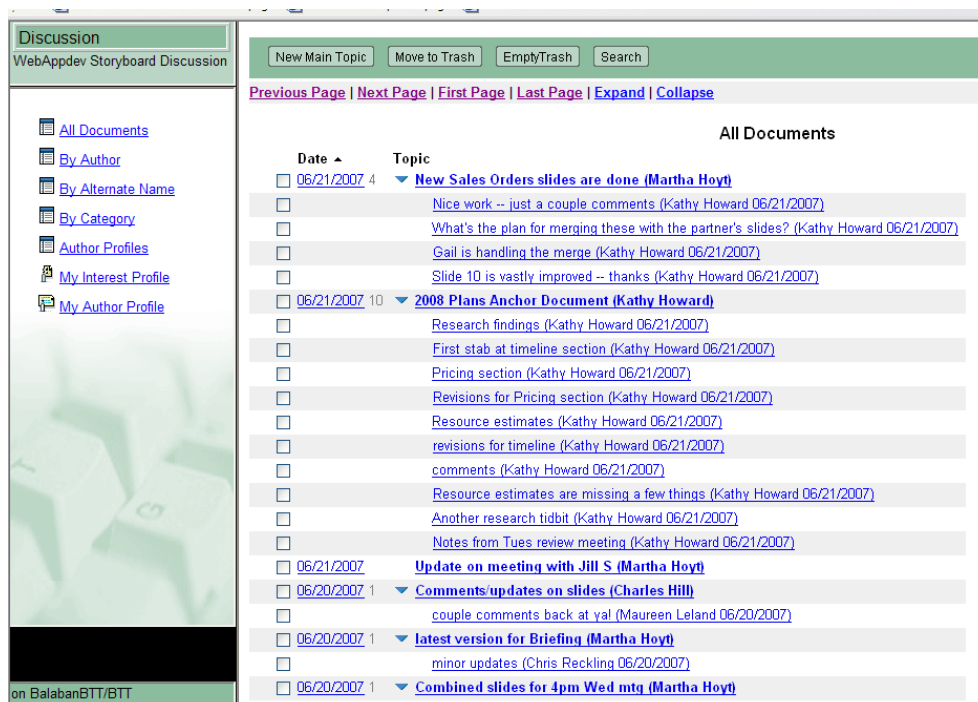
XPages jsou za poslední dobu největší změnou ve způsobu vývoje aplikací Lotus Notes. Přináší nové technologie, které předtím nebylo možné využít. Jako každý způsob vývoje mají své výhody i nevýhody.

### 5.1 Přínosy XPages

Velkou výhodou XPages je, že umožňují vytvářet webové i klientské aplikace. Stačí tak aplikaci navrhnout jednou a lze ji spouštět přes klienta i webový prohlížeč. Před XPages bylo také možné vytvářet webové rozhraní aplikací, ale nejprve se musela vytvořit celá aplikace pro klienta a poté se k ní přidávalo webové rozhraní.

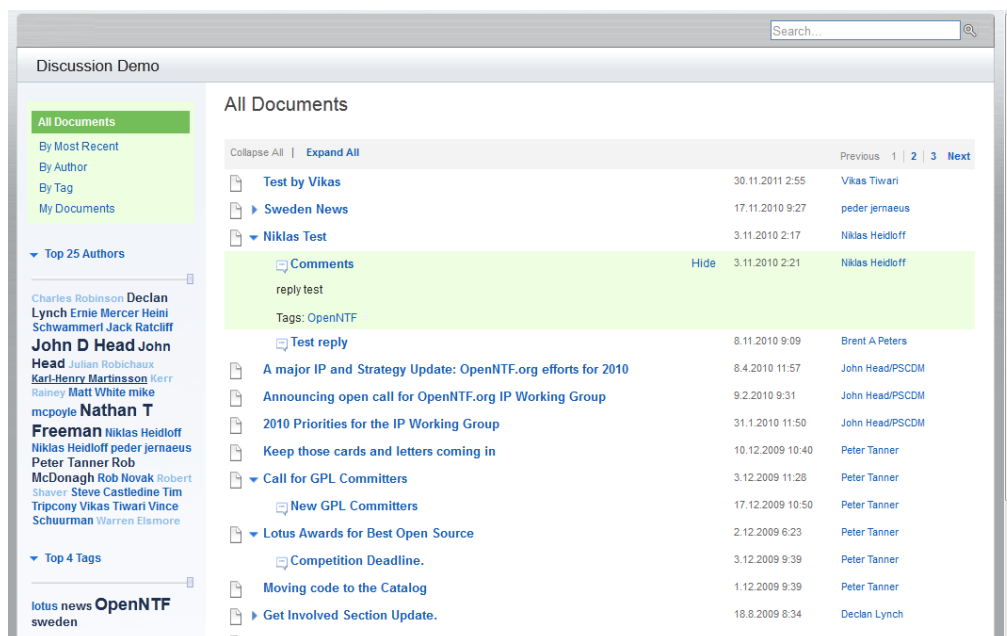
Přidávání webového rozhraní bez XPages bylo umožněno díky HTML serveru. Část formuláře automaticky přeložil HTML server a zbytek se musel doprogramovat. Kód se psal přímo do formuláře a označil se jako HTML. Využíval se k tomu hlavně jazyk HTML, dále i JavaScript, jazyk formulí a LotusScript. LotusScript se dal využít tím způsobem, že se napsal agent a v HTML se zavolal. Vzhled určovali kaskádové styly, ale poněkud složitěji, jelikož až od verze 8.5 má Domino Designer vestavěné editory pro kaskádové styly, takže se k tomu museli používat externí nástroje jako Notepad, PSPad a podobně. Tento způsob vývoje je stále možný využívat, ale XPages nabízejí grafické rozhraní pro vývoj a nové možnosti.

Pro porovnání na *Obr. 15: Vzhled starší webové aplikace* je aplikace vytvořená tímto způsobem. Stejná webová aplikace vytvořená pomocí XPages je na *Obr. 16: Ukázková aplikace vytvořená pomocí XPages*. Je vidět, že aplikace vytvořená pomocí XPages má modernější vzhled, a také přibýly nové prvky jako například tagy v levé straně aplikace. Tagy jsou klíčová slova, která pomáhají při orientaci ve webové aplikaci a při vyhledání příbuzného textu nebo obsahu.



Obr. 15: Vzhled starší webové aplikace

zdroj: Lotus Domino's XPages Workshop [prezentace]. Dallas: IBM Innovation Center, 2010.



Obr. 16: Ukázková aplikace vytvořená pomocí XPages

zdroj: printscreen webové stránky:

<http://208.85.190.65/demos/demodiscussion.nsf/allDocuments.xsp>

XPages umožňují snadný přístup k moderním webovým technologiím, kterými jsou AJAX a Dojo. S vestavěnou podporou AJAX se lze setkat například u mazání dokumentů. Při



potvrzení smazání dokumentů se neaktualizuje celá stránka, ale pouze pohled, ve kterém se odstranil dokument. Umožněno je to díky částečné aktualizaci pomocí AJAX.

Úplnou novinkou jsou také témata umožňující rychle měnit podobu aplikace, či pomocí nich zachovat konsistentní vzhled mezi aplikacemi. Existují také vestavěná témata od IBM, která podporují takzvané OneUI - One User Interface, což je jednotné uživatelské prostředí.

Pomocí XPages se vytvářejí i aplikace pro SmartPhony. Což je označení pro chytré mobilní telefony. Aplikace XPages se dají v telefonu zobrazit i pomocí webového prohlížeče, ale obrazovky jsou tak malé, že se s aplikací těžko pracuje. Proto se pro ně vytváří zcela nová aplikace pomocí mobilních ovládacích prvků, které jsou dostupné v rozšiřujících knihovnách. Aplikace jsou zjednodušené, aby se zmenšilo jejich rozlišení. Pro jednotlivé operační systémy mobilních telefonů, lze aplikace vytvořit, tak aby si přebíraly vzhled konkrétního mobilního operačního systému, a vypadají pak stejně jako jakékoliv jiné aplikace v telefonu.

## 5.2 Nevýhody

Nevýhodou vývoje pro webové prohlížeče je jejich nekompatibilita. XPages se s tím vyrovnávají dobře. Fungují na většině webových prohlížečů. Pouze se musí počítat s tím, že Internet Explorer se nechová standardně jako ostatní prohlížeče, ale s tím se setká každý vývojář webových stránek, ať už je vyvíjí jakýmkoliv způsobem.

Při vývoji obou aplikací popsaných v práci se objevil pouze jeden problém s různými prohlížeči. V pohledu u aplikace Obchodní aktivity se v druhém sloupci vypočítávají ikony k dokumentům. Ty jsou v některých řádcích prázdné. Původně v JavaScriptu nebyli prázdné hodnoty ošetřené a v Mozile Firefox se zobrazili dobře. Při testování se, ale přišlo na to, že v ostatních prohlížečích se prázdné hodnoty zobrazili jako červené křížky. Tak se musel JavaScript doplnit o ošetření při zobrazení prázdných hodnot, a pak už se ikony zobrazovaly ve všech prohlížečích správně.

Nevýhodou XPages je jejich časová náročnost. I v případě, že potřebujeme vytvořit pouze webové rozhraní ke stávající aplikaci, je čas pro vývoj téměř stejný, jako kdyby se vytvářela nová aplikace. Proto je vývoj XPages aplikací relativně drahý. Méně náročný návrh je v případě, že by se aplikace přepsala pouze do formy read-only. Aplikace se tak ve webovém prohlížeči pouze zobrazí s aktuálními daty, které se vytvářejí v aplikaci určené pro klienta Lotus Notes. V tomto případě má aplikace pouze informační charakter. Její vývoj je kratší, protože se nemusí programovat jednotlivé funkcionality, které v klientské aplikaci jsou. Na druhou stranu to není plnohodnotná aplikace.

## Závěr

Závěrem chci zhodnotit výsledky dosažené v bakalářské práci. Přínosem této práce je uvedení do problematiky vývoje webových aplikací pomocí XPages. V první části je představeno prostředí Lotus Notes/Domino, jako platforma pro vývoj aplikací. Teoreticky je popsán návrh aplikací pro klienta i webový prohlížeč společně se souvisejícími technologiemi. Vypsány jsou také základní rozdíly při vytváření klientských a webových aplikací. V praktické části je ukázán převod stávajících a vývoj nových aplikací pomocí XPages. V poslední kapitole jsou sepsány nejdůležitější přínosy technologie XPages a také jejich nevýhody.

Došla jsem k závěru, že technologie XPages je i přes některé své nevýhody dobrým krokem pro oživení vývoje aplikací. Do budoucna se dá předpokládat, že aplikací vytvořených pomocí XPages bude přibývat a klasických ubývat. S tím bude souviset větší využití webových prohlížečů jako klientů.

V tomto závěru mě utvrdil i seminář Lotus Notes, kterého jsem měla možnost se zúčastnit. Pořádán byl společností OR-CZ spol. s r.o., a prezentovali na něm i zástupci IBM. Mluvili o nových produktech, které se chystají i v souvislosti s Lotus Notes. Připravuje se nová verze, která změní klienta Lotus Notes. Přistupovat by se k němu mělo právě pomocí webového prohlížeče skrze Integrated Home Page. Což by se dalo označit za webový portál pro přístup k základním aplikacím Lotus Notes. Klient Lotus Notes bude stále funkční, ale uživatelé se budou moct rozhodnout, zda budou k aplikacím chtít přistupovat skrze klienta, nebo přes webový prohlížeč.

## Citace

- [1] *Seznam produktů IBM Lotus* [online]. [cit. 1. leden 2012] Dostupné z: <http://www-142.ibm.com/software/products/cz/cs/atoz/lotus?pgel=lnav>
- [2] KUČEROVÁ, M., MORAVEC, L. *Lotus Notes - Podrobná uživatelská příručka pro verze 8 a 8.5*, Brno: Computer Press, 2009. 400s. ISBN 978-80-251-2538-0.
- [3] BIRTOVÁ, I. Řízení pracovních toků prostřednictvím workflow, *Computerworld*. Praha: IDG CzechRepublic,a.s., 2012. roč. 23, č. 7, s 32. ISSN 1210-9924
- [4] BENZ, B., ROCKY, O. *Mistrovství v programování Lotus Notes a Domino*, Brno: Computer Press, 2005. 966s. ISBN 80-251-0750-7
- [5] *The History of Notes and Domino* [online]. 20. prosince 2005. [cit. 1. leden 2012] Dostupné z: <https://www.ibm.com/developerworks/lotus/library/ls-NDHistory/>
- [6] *What's new in IBM Lotus Notes and Domino V8* [online]. 20. březen 2007. [březen 2012] Dostupné z: <http://www.ibm.com/developerworks/lotus/library/notes8-new/>
- [7] *Eclipse Platform Overview* [online]. [cit. 10. březen 2012]. Dostupné z: <http://www.eclipse.org/eclipse/eclipse-charter.php>
- [8] KREJCÁREK, J. *Domino Designer 8.5x* [prezentace]. 5.11. 2009.
- [9] DONNELLY, M., MCGUCKIN, T., WALLACE, M. *Mastering XPages: A Step-by-Step Guide to XPages Application Development and the XSP Language*, IBM Press, 2011. 784 s. ISBN-10 0132486318. ISBN-13 978-0132486316.
- [10] *Web 2.0/Enterprise 2.0*, *Computerworld*. Praha: IDG CzechRepublic,a.s., 2011. roč. 22, č. 10, s 4. ISSN 1210-9924
- [11] GRUSOVÁ, L. *XML pro úplné začátečníky*, Praha: Computer Press, 2002. 196s. ISBN 80-7226-697-7

[12] BRIZANERA, B., BUCICA, M., DARIE, C., CHRECHES-TOSA, F. *AJAX a PHP tvoříme interaktivní webové aplikace profesionálně*, Brno: Zoner Press, 2006. 320s. ISBN 80-86815-47-1

[13] *Lotus Domino's XPages Workshop* [prezentace]. Dallas: IBM Innovation Center, 2010.

[14] BROWN, K., BROWN, K., BROWN, K., KOUTCHOUK, F. *Mistrovství v Lotus Notes 4*, Brno: Computer Press, 1997. 986s. ISBN 80-7226-011-1

[15] Příručka *OR-INFO: TRADE-INFO* [online]. [cit. 10.březen 2012]. Dostupné z: [http://www.orcz.cz/www/www-new.nsf/OR-INFO\\_TRADE-INFO.pdf](http://www.orcz.cz/www/www-new.nsf/OR-INFO_TRADE-INFO.pdf)

## Bibliografie

BERGLAND, J., DAVID, P., DENIAU, J., JAIN, A., LILL, B., SABIR, M. *Redbooks Wiki: Building Domino Web Applications using Domino 8.5.1*, verze 8. 7. června 2011.

Dostupné z: <http://www->

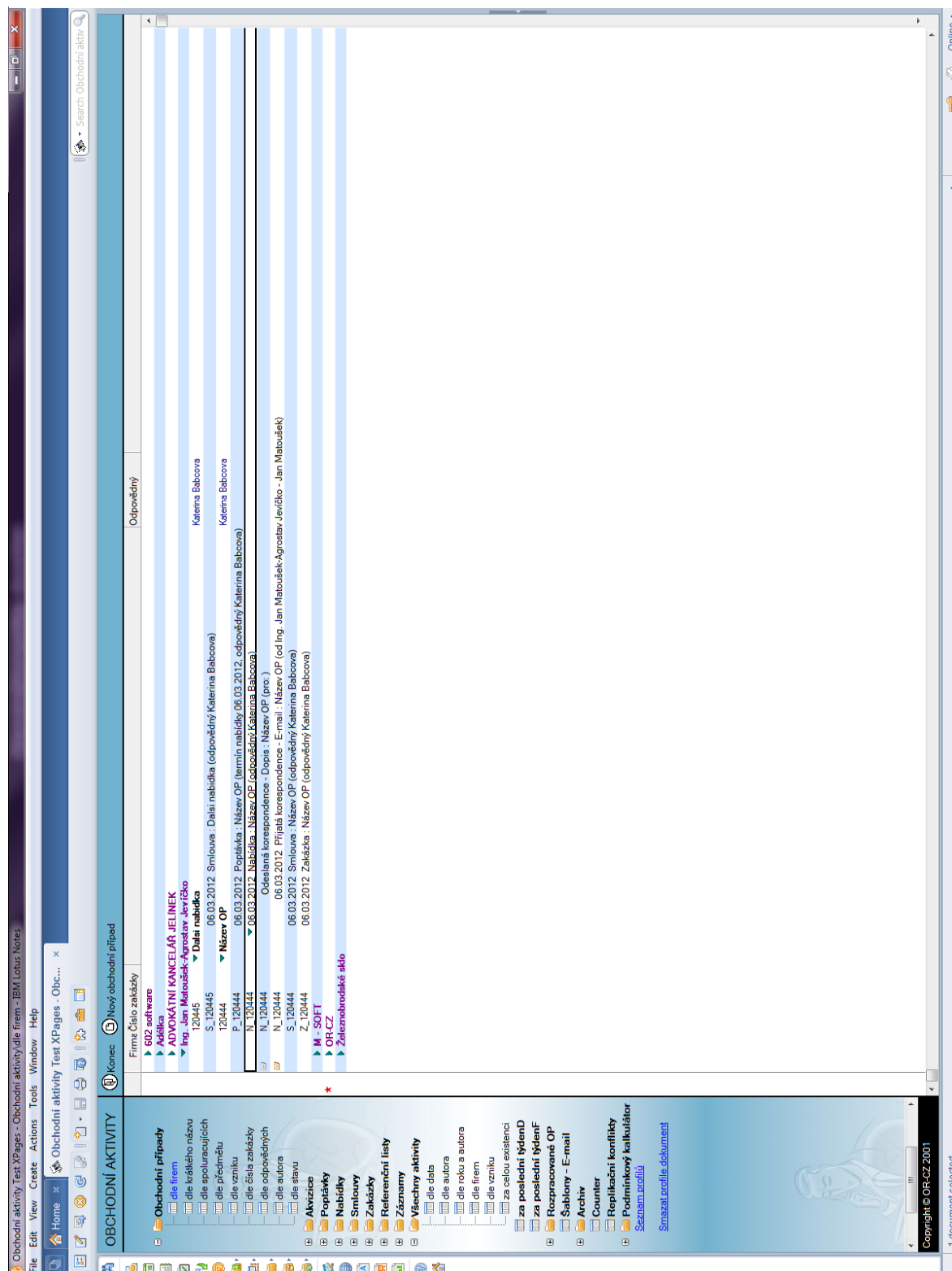
[10.lotus.com/ldd/ddwiki.nsf/dx/PDF\\_Version\\_of\\_Redbooks\\_Wiki\\_Building\\_Web\\_Applications\\_with\\_Domino\\_8.5](http://10.lotus.com/ldd/ddwiki.nsf/dx/PDF_Version_of_Redbooks_Wiki_Building_Web_Applications_with_Domino_8.5)

CASTLE, J., DAVIS, C., LIEDER, S., MONSON, P., NOSTRAME, F., OGUNBODEDE, O., SHARMA R. *IBM Lotus Notes and Domino 8 Deployment Guide* [online]. International Technical Support Organization, 2007. 504 s. ISBN 073848895X. Dostupné z: <http://www.redbooks.ibm.com/redbooks/pdfs/sg247506.pdf>

## Seznam příloh

Příloha A: Aplikace Obchodní aktivity .....	63
Příloha B: Aplikace Profily .....	67

## Příloha A: Aplikace Obchodní aktivity



Obr. A1: Úvodní stránka aplikace Obchodní aktivity pro klienta

Katerina Babcova | Odhlásit

Obchodní aktivity Test XPages

Obchodní případy  
dle firm  
dle krátkého názvu  
dle předmětu  
dle vzniku  
dle čísla zakázky  
dle odpovědných  
dle autora  
dle spolupracujících  
dle stavu

Akvizice

Poptávky

Nabídky

Smíouvy

Zakázky

Referenční listy

Záznamy

Všechny aktivity

Obchodní případ - dle firm

Firma

Účetní zakázka

602 software

Adélka

ADVOKÁTNÍ KANCELÁŘ JELÍNEK

Ing. Jan Matoušek-Agrostav Jevíčko

120445

Dalsi nabídka

120444

Název OP

P\_120444

06.03.2012 Poptávka : Název OP (termín nabídky 06.03.2012, odpovědný Katerina Babcova)

N\_120444

06.03.2012 Nabídka : Název OP (odpovědný Katerina Babcova)

N\_120444

Odeslaná korespondence - Dopis : Název OP (pro : )

N\_120444

06.03.2012 Přijatá korespondence - E-mail : Název OP (od Ing. Jan Matoušek-Agrostav Jevíčko - Jan Matoušek)

S\_120444

06.03.2012 Smíouva : Název OP (odpovědný Katerina Babcova)

Z\_120444

06.03.2012 Zakázka : Název OP (odpovědný Katerina Babcova)

M - SOFT

OR-CZ

Železnobrodské sklo

Katerina Babcova

Babcova

Katerina Babcova

Obr. A2: Úvodní stránka aplikace Obchodní aktivity vytvořená pomocí XPages

64



Formulář Obchodní případ č. OBP-2012-0447 - IBM Lotus Notes  
File Edit View Create Actions Tools Window Help

Home x Obchodní případy Test XPages - Obch... x  
Search All Mail

Obchodní případ  
Název OP  
Vývoři  
Katerina Babcova  
06.03.2012 11:11:20

Ing. Jan Matoušek-Agrostav Jevíčko (Olomoucká 226, 569 43 Jevíčko)  
Odpovědný: Katerina Babcova  
Zúčastnění: Katerina Babcova  
Číslo OBP-2012-0447

Číslo zakázky: 120444

Účetní zakázka:  
Výkazy: ☐ Ne

Aktuální dokument Dokument v archivu

STAV: **Poplávka**

Základní údaje | Kontakty | Poznámky | Další údaje a soubory | Úkoly | Vývojový strom

ZÁKLADNÍ ÚDAJE	
Název:	Název OP
Cena (bez DPH):	10 000,00 EUR
Úmístění:	20%
Předpokládaný termín realizace:	10.03.2012 14
Přednáš OP:	Docházka - implementace

Stupeň utajení: Všeobecný  
☒ Důvěrné  
Rozšířený přístup k dokumentu:

Poslední úpravy: Katerina Babcova 06.03.2012 11:13:31

Online

Obr. A3: Obchodní případ v aplikaci pro klienta

Zavřít

Obchodní případ

Název OP

Vytvořil

Katerina Babcova

06.03.2012 11:11:20

Ing. Jan Matoušek-Agrostav Jevíčko (Olomoucká 226, 569 43 Jevíčko)

Odpovědný: Katerina Babcova

Zúčastnění: Katerina Babcova

Číslo: OBP-2012-0447

Číslo zakázky: 120444

Účetní zakázka: Výkazy: Ano

Aktuální dokument

STAV: **Poptávka**

Základní údaje

Poznámky

Další údaje a soubory

Úkoly

Vývojový strom

ZÁKLADNÍ ÚDAJE

Název:	Název OP
Cena:	10000 EUR
Úspěšnost:	20 %
Předpokládaný termín realizace:	10.3.2012
Předmět OP:	Docházka – implementace

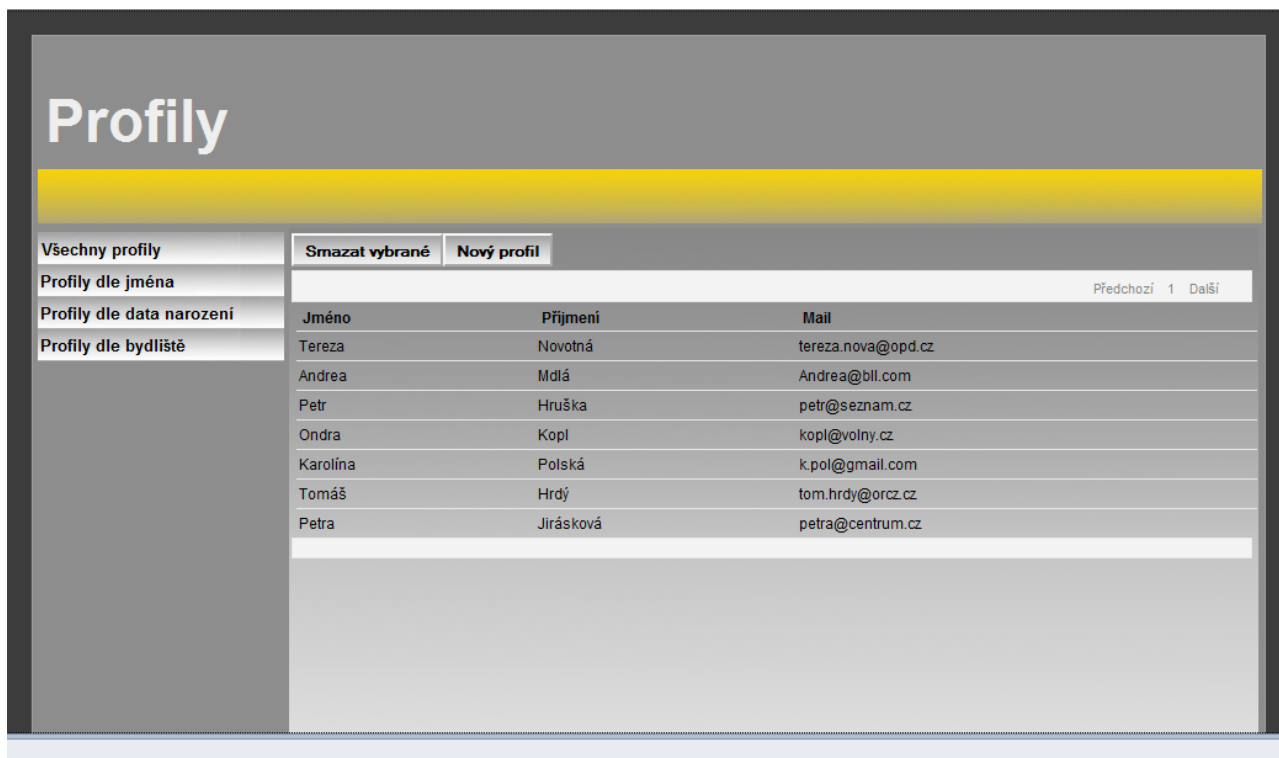
Stupeň utajení: Všeobecný

☒ Důvěrné

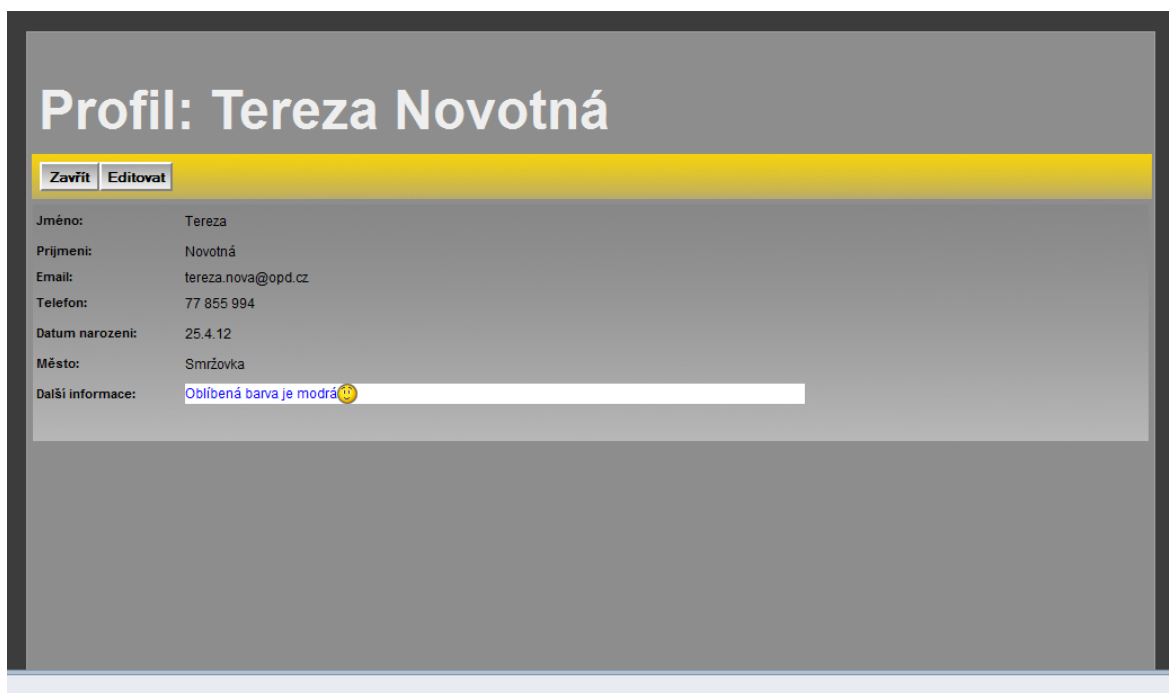
Poslední úpravy: Katerina Babcova 06.03.2012 11:13:31

Obr. A4: Obchodní případ vytvořený pomocí XPages

## Příloha B: Aplikace Profily



Obr. B1: Úvodní stránka aplikace Profily



Obr. B2: Profil